The Australian Apple Review

A Gareth Powell Magazine

The new Apple Macintosh – the future Apple as a Computer Terminal Logo – the cry of the Turtle Add-ons with Bells and Whistles Numeric keypad Disk drive analyser Buglettes Program a Winner at the Track Worm in the Apple News Letters The new Apple Macintosh – the future Apple as a Computer Terminal Logo – the cry of the Turtle Add-ons with Bells and Whistles Numeric keypad Disk drive analyser Buglett

Contents

From this issue we go monthly. They say that the third issue of a magazine is the most important. The first issue you can get away with murder, because you're just starting out. The second issue there's always room for improvement. But the third issue: that's when you've got to consolidate and prove that you're not just a flash in the pan.

We have plenty more items of interest this month. We have an in-depth look at some of the different communications packages available for the Apple II. "Electronic Communications" is a bit overrated, but it definitely has a place. We also continue our look at computer languages available for the Apple, with the popular educational graphics language "Logo". In future issues we will be examining Pascal and FORTRAN.

Unless we are truly flooded with boring and uncomplimentary letters, we would like to publish as many of your letters as possible. This is your magazine, a place where your views can be exchanged.

A number of people have asked us about writing for the magazine. We welcome suitable articles, but being a new magazine we have a restricted editorial budget, practically all of which goes to pay our professional staff writers and regular columnists.

Consequently we can offer little more than the immortality of seeing your name in print, and a free subscription to the magazine, to budding authors. If that's enough, get in touch with us. We want good general interest articles, preferably not too technical. How useful they are to us depends a lot on that factor, and on how well they are written. (It's amazing how many people think that they can't write but they can.) As the magazine grows, so will the amount which we can pay contributors, but bear with us for now if you're looking to break into the wonderful world of computer journalism.
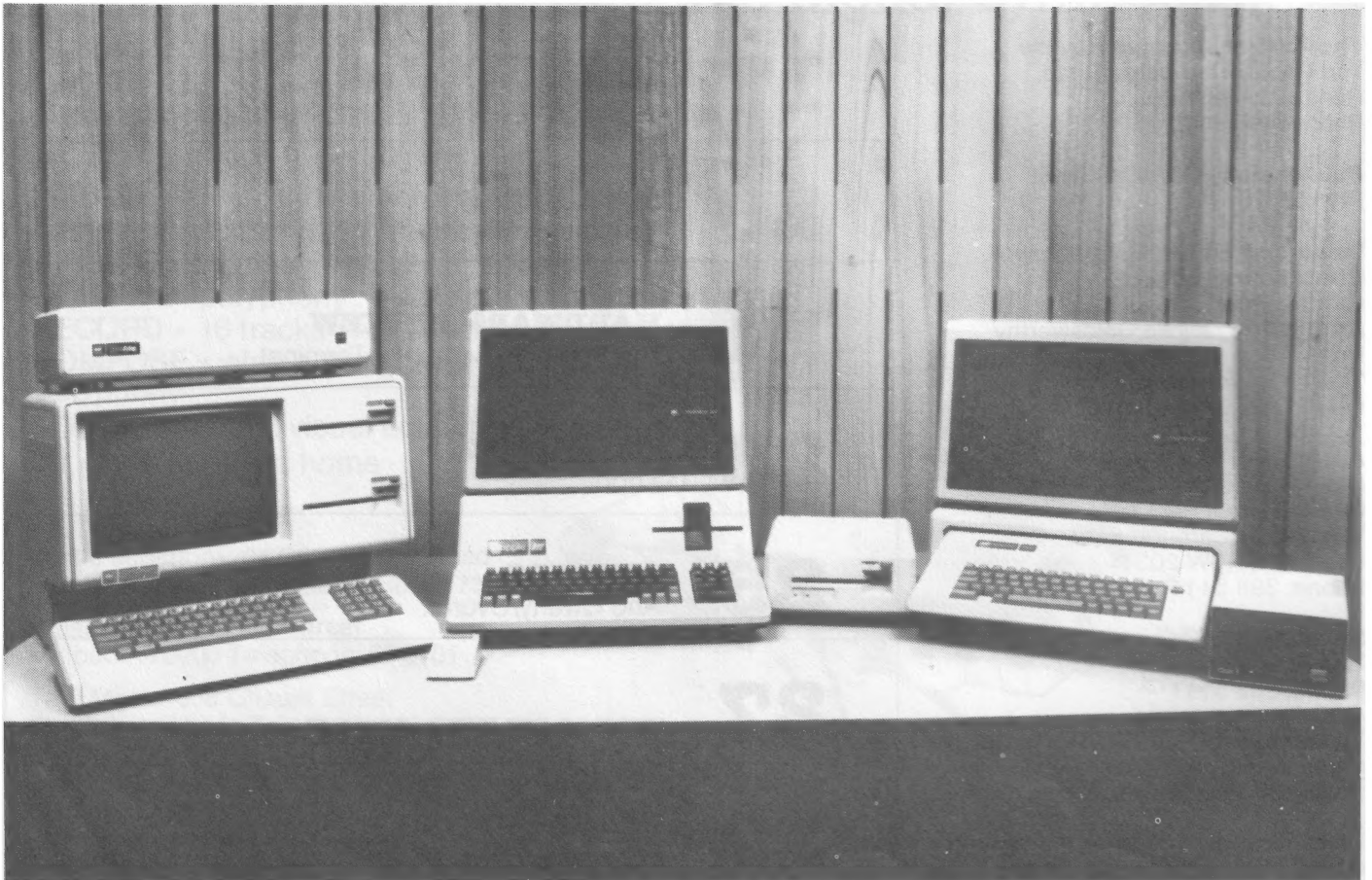
Macintosh order of magnitude

Peat Marwick and Mitchell, who go around auditing accounts to see that they have not been fiddled, have ordered 2,500 Apple Macintoshes. That order has set them back something over US$7,000,000 and they will be using their own software for their auditors (of which there are 7,000) to indulge in communication, analysis and financial modelling.

New big time disk makers

Elsewhere in this issue we refer to the new rainbow coloured discs. This presages the start of a colourful time in microcomputers because both Kodak and Polaroid have now announced they are getting into the floppy disk business. When you come to think of it they have left it a little late. By the time they get into the swing of things we will probably be out of disk drives and into some form of gigabyte memory in a small box powered by a battery. Nevertheless, with these two giants easing their way into the market we can at least bet that the packaging will be colourful.

Apple software developers conference

More than 50 developers of software, including Felix Macri from this magazine, fronted up for the first Australasian Independent Software

An Apple by any other name

We are sure all that our readers are aware of this but for anyone who does not yet know, Macintosh is named after an American type of Apple. Not after the type of raincoat which is used by dirty old men for the purposes of flashing. Now, are you quite clear on that point? We are pleased they didn't call it Granny Smith.

Awesome AUSOM

The Apple Users' Society of Melbourne (membership enquiries to Grahame Willis on (03) 878 0219 or David Halprin on (03) 387 3221) publishes its own magazine called, understandably, "Ausom News". The last issue contained two full page advertisements for machines which we will refer to politely as "Apple Compatible". It also announced the formation of a Medfly (a machine that will run Apple software) group. In searching for suitable acronyms for a title they have come up with MEDUSA (Medfly Users Subgroup of AUSOM). Where will it all end?

Developers' Conference held in Sydney last month. Organised by Apple, the conference was talked to by several software developers from the United States and Australia. Felix will be reporting in full on this conference in next month's issue.

Japan sorts out copyright

While Australia still has not worked out the copyright situation on software, Japan steams ahead to get the situation defined and legislated into some sort of order. Japan's all-powerful Ministry of International Trade and Industry will have the draft bill ready in a matter of weeks. Under the proposed Japanese legislation it will be illegal to use or copy software without the permission of the developer. In return, the developer will have to register the software in order to preserve these rights. The legislation will cover both the object and source codes but not the documentation, flow charts and output. This is fairly obvious as the current copyright law in most countries protects any material that is published. And in this context to "publish" merely means to write it down on a piece of paper and hand it to someone.

Under the proposed legislation the period of protection will follow patent law and be for fifteen years rather than the life plus fifty years which is afforded under normal copyright laws.

When Australian Attorney General Gareth Evans (no relation) eventually brings down the new Australian legislation it is suggested that it will retroactive. Apart from the fact that such retroactivity makes the law into a bigger ass than it normally is, we wonder how exactly the Government will go about back-dating charges. If it is anything like their handling of the retroactive legislation on "bottom of the harbour" tax avoidance schemes, it will be a right bog-up and an extra Christmas for the lawyers.
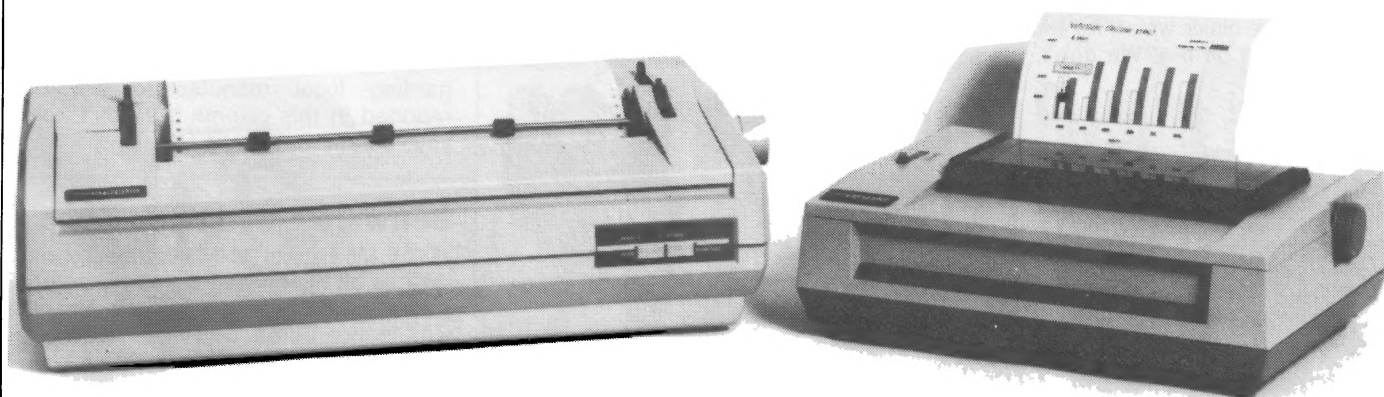
D Manual Controller

On this magazine we are not stupider than your average hacker. We're not brighter either. Sort of middling.

From CompuMusic came a D manual controller to test. The instructions start "How many times have you been working with your Apple and had to look up control addresses to select HIRES-NO MIX or LORES-MIX?" The simple answer is never. So we phoned up Ken Guntar of CompuMusic who is importing the board and asked in all innocence, what it did. Ken told us that he had no idea either but he knew it was important and that all serious programmers would need one.

The information sheet is set in a new type face, 5 point Gibberish and is written by someone who is plainly two sandwiches short of a picnic. But if this is the sort of card you are looking for — this is the sort of card you are looking for. And may we be the first to congratulate you?

Slot City

The Apple II has an abundance of slots compared to the IBM PC which only has four. But even with the Apple you can easily run out of slots. For example in slot #0 let us put 16K of expansion, in slot #1 our dot matrix printer, slot #2 our RS 232-C communication card to the Wang word processor, slot #3 the Vision 80 card and so on. Coming to the rescue is "Switch-A-Slot" which gives you three extra slots. (There are four slots in the device but you take up one slot on the board plugging it into the Apple). It only draws power for the board that is being used. The makers admit that the system sometimes has problems with Z 80 cards. We have news for them. Everyone has problems with Z 80 cards.

The box is put together by the Southern Californian Research Group (who are also responsible for the rest of the products in this sub-set) and feels as solid as a biscuit tin. The documentation that accompanies it inspires no confidence. They call them "preliminary instructions". We'll wait for the hard-back version.

Ken Guntar is also importing some technicolour disks which he thinks will set the world on fire. We don't, but he may well be right, we may well be wrong. We often are.

For those of us who play games with joysticks and paddles comes "Paddle Adapple" which allows you to use four joysticks at once. As far as we can ascertain the whole box and dice lies outside the machine in a nasty tangle of wires and happy home experimenter electronics as featured on special at Dick Smith's. Not, we think, for us.

Finally, a keyboard adapter which allows you to switch from the traditional QWERTY to Dvorak (mentioned else-where in this issue). This is a good idea, especially as it can also be used for providing one handed keyboards for the handicapped. Once again the instructions give a feeling of insecurity and doubt. They are full of spelling mistakes and the device itself looks like part of the "Electronics — It's Easy" kit from ETI.

As it happens this is a bloody good idea which, in fact, we have managed to achieve through software. Skiting aside, the concept is excellent and well overdue.

What the Southern Californian Research Group needs is some packaging, manual writing and design which live up to the pretensions of their title.

Australian Apple

The managing director of Apple Australia is reported as saying that Apple may be manufacturing in Australia by the middle of this year. As IBM have announced that they will be manufacturing (assembling is perhaps a more accurate term), their PCs in Wangaratta, Victoria, this seems a logical move. If IBM moves can Apple be far behind?

This news about Apple Australia's intentions lacks interior logic.

According to the statement the only machine involved will be the Apple IIe. Plainly it won't be the Macintosh — that is manufactured in an almost totally automated factory in Fremont which cost a bundle and produces a thousand machines every shift. But it is also reported that production levels would have to be well above 1,000 a month to make the project viable. David Strong, the managing director of Apple Australia is quoted as saying "We would try to turn out whole machines rather than just components but whether we would go into disk drives is still to be determined".

This would mean that the Apple IIe would have to sell in Australia over 12,000 units a year. That is so very far ahead of anyone's sales forecasts. Bear in mind that there is already an assembly facility in Singapore and it is unlikely that there will be an export market for these machines. If the total number of Apples sold so far in Australia is about 20,000 (a figure which was quoted in legal proceedings) then how can anyone expect the Apple IIe to accelerate to 12,000 units a year? The whole promotional thrust of Apple in the next year will be centred around the Macintosh, expected by Apple to provide 35 per cent of the company's revenues. Further developments regarding local manufacture will be reported in this column but don't hold your breath.

Software Liberation

Will Mr Langer of Melbourne please take us off his mailing list as we do not wish to give exposure to his company as we do not agree with their business methods.
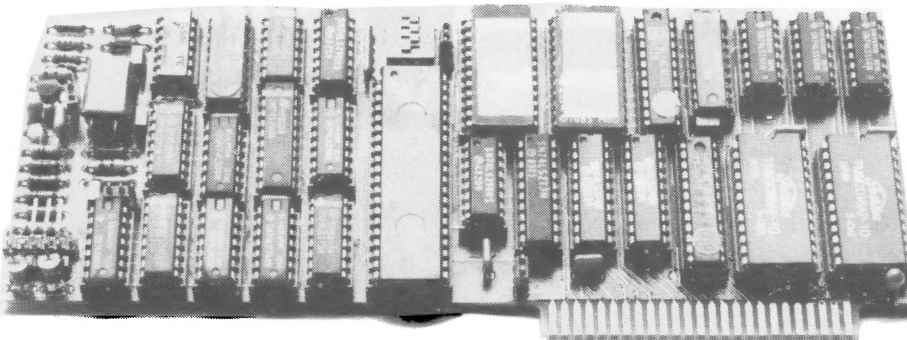
The screen of the Apple Lisa

The new Apple Macintosh
"I have seen the future – and it works"

by Gareth Powell

The Macintosh main unit with keyboard and mouse

Before I go in to a review of the Macintosh, which I have been using as a personal computer for some time, it might be appropriate to air some views on Apple management and their approach to their own product.

There seems to be a terrible compulsion in Apple headquarters to bring the IBM PC into the discussion every time the Macintosh and the Lisa are mentioned. Why this should be I cannot imagine unless it is a case of corporate paranoia. "Just because we are paranoid doesn't mean that the rest of the world isn't out there to get us."

There is absolutely no valid comparison between the Lisa/Macintosh technology and the IBM PC. In fact, a fair case could be made that there is no competition anywhere in the market place.

The IBM PC is a stolid machine which has been made to assure middle level business management that microcomputers will not turn and bite them on the bum if left unattended. They are not intended to be the state of the art. These are not our own original thoughts. They are the pronounced dicta of IBM management.

One senior executive of IBM recently said in public "IBM does not want to be on the leading edge of microcomputers". IBM are, for anyone's money, the best marketing team in the world. The moment they announced their PC there was little doubt that they could, would and did dominate the market that they selected. But it is a very special market and there is little or no crossover between that machine — a stolid, unimaginative but reliable business tool with the security of the three famous initials on the front — and the state of the art pyrotechnics of Lisa/Macintosh technology.

And the sooner that this is burned into the subconsiousness of all Apple employees the sooner they will be able to correctly and intelligenty identify their proper markets.

Amazing Mac

Having got that little lot off my chest let battle commence. The Macintosh is

one of the most amazing computers I have ever used.

It is a natural extension of the Lisa. It is a microcomputer with facilities which, frankly, I had not dreamed about.

Of course, there are several other facilities that I would like and I will list them later. But for an overall view suffice it to say that I consider the Macintosh is in almost every respect the definition of what a personal computer should be — from design to documentation, from concept to finished product.

When I wrote my first review of the Macintosh for the "Weekend Australian" I asked the question "Who is it for? Who is going to buy it?" I think I now know the answer to those questions.

Concept

It is not generally accepted at Apple headquarters but the Macintosh might well have been launched as the Lisa junior if it hadn't been for the monumental bog-up that they had made in launching the Lisa.

Now they have got the Lisa sorted out into what is, in my opinion, the best value-for-money and the most desirable microcomputer on the market today, it is logical to think of the Lisa as the top end of a family of computers and the Macintosh as the other end.

Individual parts

Looking at the Macintosh in its individual parts: there is a main unit, a keyboard and a mouse.

In the main unit is the processor, the memory, the built-in disk drive and the screen.

The processor is a 32 bit MC68000 with a 7.8336 Megaherz clock frequency.

The memory is 128K Bytes RAM and 64K bytes ROM. Although it does not say so in the literature or any of the press handouts there is no doubt that an expansion chip will shortly be available for the RAM. Almost certainly this will be a 256K chip bringng the RAM memory up to a very respectable 384K which, for certain applications, is necessary.

The built-in disk drive is a Sony and uses 3½ inch disks in hard cases which provide far more protection than the proverbial floppies. Apple's publicity machine keeps telling us that these disks will fit in a shirt pocket and a quick test shows that indeed they do fit into a shirt pocket. A more hostile environment for a disk is hard to imagine. But if you want to walk around in the heat of the noonday sun, it is true you can keep your disk in a shirt pocket that is soaked with perspiration.

400K formatted

Each disk holds 400K when it is formatted. Eventually, we will be able to get rid of disk drives altogether and have a few gigabytes of memory stuck in

You push the mouse around on your desk to move the little arrow

something rather smaller than a shoe box running off a couple of torch batteries. Until that happy day arrives these disks will do very well.

The screen is 12 centimetres on the diagonal and has a 512 × 342 pixel display. It is important that the size and resolution of this display be remembered as it has a great bearing on some aspects of the software which we will investigate later on. It does not have colour but is in the full glory of black and white.

At the front of the main unit is a American telephone type socket to receive the switchboard jack. At the back are a series of interconnectors which are extremely clearly labelled. There are two RS232/RS422 serial ports which can handle a maximum of 230,400,000 baud, or more readably 230.4K baud.

Transmitting and receiving data at 9.6K baud is currently regarded as in the Superman class. The fact that speeds of up to 230.4K baud are theoretically acceptable suggests that in communications there is a new day adawning. How the hell they are going to transmit at

these speeds from an external source befogs my simple Welsh peasant's brain but no doubt all these thing will be explained to me in the near future.

There is a second disk drive interface and a little hole for the mouse to hang on to. The Macintosh is a microcomputer which is totally dependent on "mouse" technology.

Timorous beastie

If you do not love using a "mouse" then the Macintosh is not for you as there is no way you can operate the machine successfully without using a "mouse".

Inside the main body of the church is a sound generator and a clock/calendar operating from a battery which is user-replaceable. We mention the "user-replaceable" aspect with serious intent because, as far as we can ascertain, that is the only part of the Macintosh body that you can open without making your warranty null and void. This machine is not designed for the happy home enthusiast who wants to insert bits and bobs and then paints on go-

faster stripes. Not that this will stop anyone. It will be interesting to see who will be the first manufacturer off the mark with an improvement for the Macintosh.

My guess is that it will be someone who makes a more intelligent mouse.

The mouse, incidentally, is the only other piece of user-serviceable equipment attached to the machine. You can partially dismantled it to clean the rotating ball on its underside (this mouse, like Hitler, has only got one ball) where it has picked up the sticky mess which is always on the surface of any working person's desk. It is an operation that can be performed with sadistic delight as it is the only time you feel that a human has full control over this wretched creature.

The Macintosh is a small machine and when you first look at it you tend to think it is ugly. After a while you realise that form + function = style and you begin to appreciate its aesthetics. At least I did.

No manuals needed

The machine has been designed to be

The back of the Macintosh
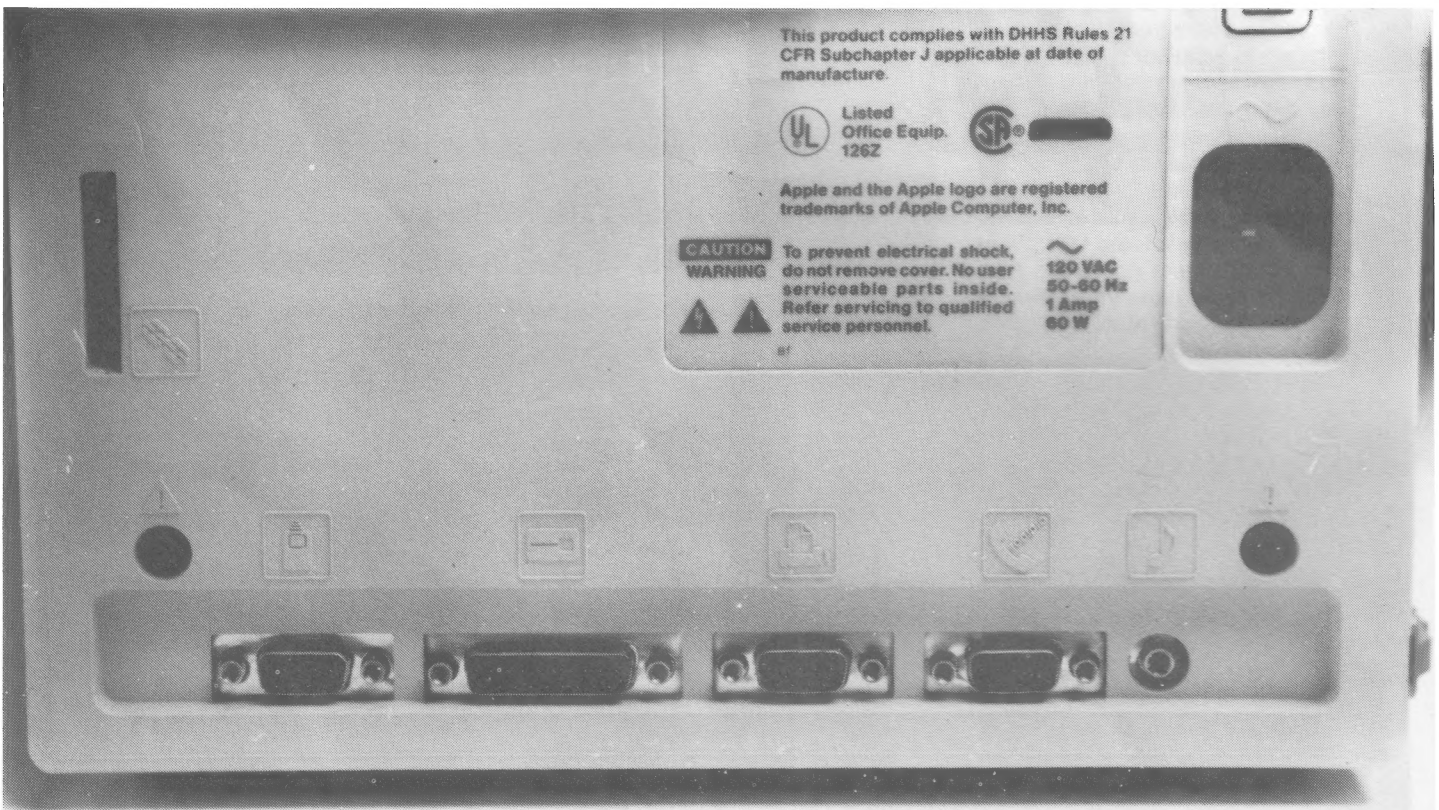
Screen and disk slot

used with the least possible need for reference to training manuals. This is despite the fact that the main manual is one of the best designed, best written manuals I have ever seen, albeit sexist to a grossly offensive extent. It is illustrated as if there were no women, no negroes and no people over the age of thirty anywhere in the world. A pox take the manual designer.

Extensive use has been made of little drawings on the screen which Apple have decided to call icons which can be used, according to my dictionary, in both the normally accepted religious sense and to mean an ordinary image or picture. Apple are grammatically correct in their designation so icons it shall be.

You see the very first icon when you switch on. If you have the right disk in

the drive there is a drawing, excuse me, icon, showing a miniature Apple with a smiling face in the centre of the screen if all is well. Once the system disk is booted (which takes a comparatively short time) you see a screen with five indicators along the top left hand edge. These are, in order, a drawing of an Apple, "File", "Edit", "View", "Special". Down the right hand edge of the screen you will see icons. Elsewhere on the screen you will see a small arrow which is, in effect, a cursor showing where the mouse has got to.

It would be difficult to beat the manual for the clarity with which it explains the functions of the Macintosh. It says:

"Working with Macintosh is very different from working with other computers. You use the mouse to

point to things on the creen and to cause most of the action to happen. Usually, you only use the keyboard to type text and numbers.

Your everyday routine with the Macintosh will probably be something like this:

1. Switch the Macintosh on and insert the disk with the application you want to use and the documents you want to work on. (You can keep your Macintosh always switched on if you like; just turn down the brightness control whenever you finish working with it.)

2. Open the document you want to work on or open a brand new document by opening the application you want to use. (OK this sentence is a little obscure but read on and all will be revealed unto you).

3. Use the mouse and keyboard to create or change the document.

4. Save your work frequently by choosing the Save to File menu.

5. Use the Clipboard to move things from one document to another by choosing Cut and Paste from the Edit menu. Add pizzazz to sales reports with a picture from MacPaint, or drop the proposed budget into your department's monthly status report.

6. If you have an Apple Imagewriter printer and you want a printed copy of your document, choose Print from the File menu.

7. When you've finished working on a document, choose Save from the File menu to save your last changes, and then choose Quit from the File menu.

8. Anytime you switch the Macintosh off, the disk should be ejected."

It was in reading this summary that I started to get a glimpse of what Apple was driving at.

A working tool

The Macintosh is NOT a microcomputer as you and I think of a microcomputer. It is a working tool for the office. Just like a telephone, a calculator or a manila folder. In fact, it incorporates the last two into its small frame.

The intent, the thrust, the idea of the machine is that it should be a natural part of your desk furniture which you will reach for as easily and with as little fuss as you would reach for the telephone.

The Macintosh, despite all the bells and whistles, is a simple aid to office

efficiency. Once you get that idea firmly stuck in your mind all the other aspects of the machine fit into place.

The Macintosh is physically small because you need room on your desk for other working tools — including your morning cup of coffee. The Macintosh is not intended to be kept away in a work station of its own. Like the telephone, its place is right there on the desk.

The icons and screens are there so that you don't have to look at the manual in order to do things. Retrieving documents, amending or adding to them and filing them away is accomplished by pointing the little arrow to the right symbol. You never have to type in complex codes, you never have to use Control "T" or any other keys in order to accomplish a task. You point the arrow, press the button and it all happens.

At first I was very anti-mouse. And, in a sense, I still am because it is designed with little or no regard for ergonomics.

But now I can see why the mouse is an essential part of the Macintosh.

Without it the user would have had to use a complex array of control characters to achieve a given effect which would have defeated the whole point of the exercise — to create a simple machine that can be used almost without thinking. A machine where the user's intelligence and memory is directed to the job in hand and not to the operation of the machine.

Once I had got this basic premise straight in my skull the rest was easy. For the purpose for which it is intended the Macintosh is a design of genius.

Incomparable

It is wrong to compare it with any other microcomputer on the market. Indeed, it is a little misleading to call it a microcomputer. Car engines are often controlled by microcomputers — but they are still car engines. The Macintosh is a tool for handling information which is controlled by a microcomputer — but it is still an information tool.

This information tool has a lot of bells and whistles on it which detract from the noble simplicity of its basic design and its ease of function. As I describe these bells and whistles bear in mind that they are enhancements — in some cases distractions — to a simple information processing tool which becomes second nature to use.

Word processing

When you are using the Macintosh as a word processor you can choose from as wide a range of type faces as you can imagine — in assorted sizes and weights. In a sense the Macintosh is a fully equipped typesetting unit. This opens up a series of interesting possibilities.

First of all it enhances the machine as an information tool as it enables the user to make reports and letters look much more professional and polished with absolutely no fuss whatsoever. But it may, in the near future, go much further than that.

Typesetting future

If you take a letter up to its maximum size which is 72 point (about three centimetres tall on the screen with most type faces) you will see that the rounded curves of, say, the letter "o" are in fact a series of nasty jagged steps.

Why is this so?

Because the screen only has a limited amount of pixels to play with and therefore a true curve cannot be achieved. Neither can a true diagonal. The smaller the type size you choose the less that this will be apparent but the effect is still there.

This limitation is a limitation of the screen and is almost certainly not a limitation of the bit map inside the machine. If this is indeed so, and I am assured that I am probably right on this point, it is feasible that a driver program can be written which will take those apparently jagged edges on the screen and transfer them to a printer which will, if it is capable, make the edges round and smooth again.

Such a printer is not Apple Imagewriter which is a dot matrix printer (probably a C. Itoh 8510 in drag if past experience is anything to go by). But it could be one of the new generation of laser printers which are being released.

If the Macintosh can transmit its typed text to a laser printer through a driver which will smooth out the jaggedness imposed by the limited number of pixels on the screen then, in two years or less, we will see the end of typesetting as we know it today.

This issue of the magazine was typed on Apple disks which were then sent to a typesetter who pushed them through his phototypesetting machine. This machine made images on photographic paper which was then processed into pages of typesetting. If what I have surmised is true that will not, in the very near future, be necessary.

The type faces, size and style can be selected with consummate ease on the Macintosh. Almost every type face known to man will become quickly available on disks — and if you don't like any of those you can easily design your own. Once the copy has been prepared and the typefaces chosen — something that becomes almost a reflex action as you get to know the house style — then a print out on a laser printer could provide typesetting of a quality which will rival that of the phototypesetters. But it will be on plain paper, it will be instantly available and it will, in effect, be free.

I've gone on about this a bit because as a journalist and a publisher I believe that this new technology is going to have an immense effect on every aspect of my life. And the lives of the hundreds of thousands of other people who are involved in the information business. I truly find the possibilities awe-inspiring.

MacPaint

Less inspiring by a an order of magnitude is the program MacPaint. This is the program which has drawn all the gasps of astonishment from the privileged viewers who have seen the machine in action.

But what is it for?

True, you can doodle away, then enlarge or contract those doodles and insert them into a manuscript. But unless you are an artist the doodles will look exactly what they are, doodles. Well executed and cleverly printed doodles. But doodles nevertheless. I can do better with a pencil and paper.

On page 114 of the manual there is a drawing of a 35 mm camera which, we are meant to believe, was drawn using MacPaint. Sceptical Welsh Calvinistic Methodist that I am, I do not believe that the drawing as shown was produced by MacPaint in that size on that screen using the mouse in free hand form.

I think it is possible that some type of template existed to build up the drawing in its component parts. Or, alternatively, I am willing to believe that the drawing was assembled in sections using the MacPaint system and then fitted together like a jigsaw puzzle. But Leonardo da Vinci would not have been able to make a free hand drawing using a mouse with such perfect precision as is seen in that illustration.

Cui bono?

In law, there is an useful Latin phrase, "cui bono" — who benefits? Every time I look at a program or a device in computing I ask myself "who benefits?"

Just marginally I can see MacPaint being used by art directors for graphic roughs in advertising agencies. Just possibly I can see ways in which pre-drawn objects can be introduced through prewritten programs so that on disk you will have a library of drawings to add to your reports. In which case you will be able to make your letters and reports look pretty.

Other programs

Other programs are, of course, already available. MultiPlan is an excellent example. It has been written so that it fits in exactly to the standard format of the Macintosh and you see a screen and use commands which are intended to become second nature.

In truth, Multiplan will not be as useful as it might be until the RAM is extended because a spreadsheet takes up large amounts of memory to be really effective.

Many other programs are becoming available from MacProject which gives you Critical Path Analysis, MacTerminal for making use of the communications port and by the time the machine is released in Australia (which Apple say will be in April but I have my doubts) there should be over 200 programs available, all of which will fit into the Macintosh standard format, all of which will use some of the 400 routines that are buried in the ROM.

Conclusion

The Macintosh is an amazing machine.

If you try not to think of it as a microcomputer and try, instead, to see it as a standard piece of equipment to be used in an office then you will start to appreciate the beauty of the concept.

When I first reviewed the Macintosh I was dubious. Now, on the road to Ryde I have seen the light and I am converted.

PS. I see that everyone of the designers of the Macintosh have signed their name on the inside of the case. Somehow it reminds me of Shelley's lines "I am Ozymandias, King of Kings. Look upon my work and despair", □

When is 180 CPS faster than 200 CPS?

When you have this little beauty on hand. **Through-put speed** is what counts: C.Itoh's new low profile dot matrix printer is a high speed performer, prints out with a throughput of 100 LPM (draft mode with pica character).

You'd expect to lose visual quality? No way! C.Itoh's print resolution is sharp, defined, unbeatable — throughout the print head's 100 million-plus character life.

8510S is rugged, sturdy as it is fast, built to take the punishment with heavy duty castings and high quality metal parts throughout. Plastic parts are all high stress.

It's a colourful character: there's even a colour model which gives you up to seven colours, with high resolution graphics of 144 x 160 dots per square inch on all models.

And if you want to use all the features of your IBM personal computer, C.Itoh has two special versions designed just for the IBM PC.

So make sure you ask the **throughput** of your new printer. And take a close look at C.Itoh. It answers all the questions.

C.ITOH

Add-ons with bells and whistles

by Graeme Philipson

"New generation" peripheral cards

New computers now being released (we won't mention any names) have made the Apple II obsolete in many ways. The newer machines are bigger and faster, they have an 80 column screen as standard, they have more standardisation, they have larger capacity disk drives.

But the Apple II is still, in its //e form, one of the world's most popular machines. Why? Because of its truly amazing adaptability. When he designed the Apple II six years ago, Steve Wozniak had the extremely good sense to include eight expansion slots, which have since been used by Apple themselves

and a host of other manufacturers to design a million and one peripheral cards. It is these cards which have enabled the Apple II to keep up with the competition.

It is still easy to boost an Apple II so its performance stands beside the newer breed of microcomputer. This is usually not the best option for someone starting out in computing, particularly business users, but it represents a definite option to those already using an Apple II and who are thinking of trading up to a larger or more sophisticated machine.

Sydney computer dealer CompuMusic (who import the AlphaSyntauri music system reviewed in this magazine last issue) have recently

begun to import four of these "new generation" peripheral cards which truly increase the power of the Apple II. They are the APPLICARD, a Z-80 computer-on-a-card which allows the Apple II to run CP/M software, the Videx UltraTerm 160 column (that's right, one hundred and sixty – count 'em) card, the 88-CARD, and the EPS keyboard.

The APPLICARD

For a long time now Microsoft have made a card for the Apple II which allows it to run CP/M software. CP/M is a very common microcomputer operating system, similar to but very different from

EPS keyboard

Apple DOS. By using this Z-80 card you were able to run a large number of CP/M programs not normally available on the Apple.

Microsoft's card was enormously popular. At one stage Apple/Microsoft Z-80 hybrid machines actually constituted the majority of CP/M computers in the world, truly a case of the cart leading the horse. But the Microsoft card has one major disadvantage; its speed. It was, and still is, terribly slow. It connected into all the normal Apple systems: memory, I/O, screen, and had to spend a lot of time just performing these functions.

The APPLICARD is a different beast altogether. It is not a slave card, it is a full Z-80 computer in its own right. If you connected up the power to it, you could actually use it as a computer. It uses the Apple as a slave: it has its own RAM, its own video drivers, its own I/O. It makes the Microsoft card look very slow, even obsolete. The normal non-CP/M Apple runs at 1 MHz, or one million cycles a second. The Microsoft card runs at twice that speed. The APPLICARD runs four to six times faster.

To make matters even more interesting, the APPLICARD is available with an EXTRA 128K of RAM, still on the board. This is done by attaching a secondary board to the front of the APPLICARD (who said peripheral cards can't have their own peripheral peripheral cards?) packed with 16K RAM chips. With this option, and this is how they are being imported into Australia, the APPLICARD is capable of some truly amazing feats. When this extra memory is used as a RAM disk, the increase in speed becomes even more phenomenal.

This vastly increased speed is especially apparent when using the APPLICARD with a program like Wordstar, which is constantly accessing its disk. Now those of you who have read previous columns of mine (hands up who was paying attention) will know that I am no fan of Wordstar's, but it is a very popular and very powerful word processing program. I never liked it on the Apple because it was very slow, and because the Apple's keyboard was too limited. The APPLICARD solves

the first problem, and for the second see the review of the EPL keyboard below.

The more you delve into the APPLICARD's secrets, the better it seems to become. You don't need to use an 80 column card, as the APPLICARD's "SoftVideo" provides a 70 column screen using the Apple's high resolution capabilities, which is all you need for word processing. It also provides for horizontal scrolling in forty columns, allowing you to view a 255 character screen through a forty column window. If you do have an eighty column card, the card supports all the popular makes (including the Vision-80, it seems, though this is not documented).

All pretty amazing, huh? I can't fault it, it does everything it claims it does and does it well. I once thought I would change to CP/M, but gave up in disgust at the slow speed. I'm still no fan of CP/M, but that's just a personal thing. If I had an Apple (which I do) and I wanted to run CP/M (which I don't), I would

looking no further than the PCPI APPLICARD.

UltraTerm

Where will it all end. Also available, from Videx (the people who brought you VideoTerm), is the display card to end all display cards. Would you believe 160 columns of text on an Apple? Neither did I till I tried it.

The UltraTerm is an innocent enough thing to look at. It looks much the same as any other 80 column card, and connects in a similar fashion: one lead to monitor, one lead to the Apple's normal video output. It will go into most slots, but slot 3 is best, because that's where Pascal expects such cards.

I unplugged my Vision-80, plugged in the UltraTerm, did a PR#3, and there I was in 80 columns. Videx claim that in normal 80 column mode the card completely emulates their successful

Software for EPS keyboard

Applicard
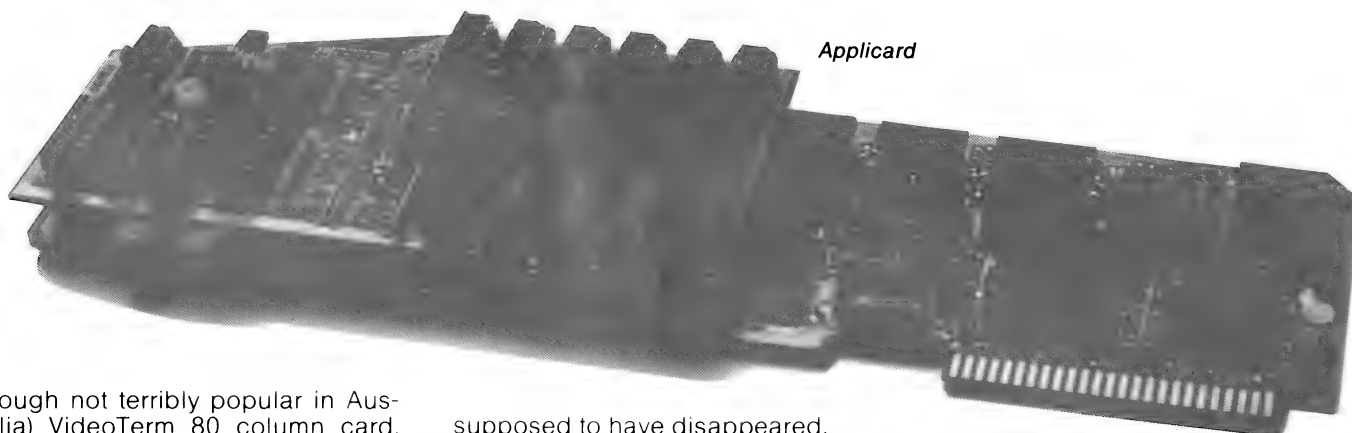
(though not terribly popular in Australia) VideoTerm 80 column card, and I see no reason to doubt this. The character set seemed quite good, better than I remember Video-Term having, but not quite as good as my beloved Vision-80.

So far it's just another 80 column card, but now the fun begins. There are nine different display formats available:

- Normal Apple (40 columns x 24 lines)
- VideoTerm emulation (80 x 24)
- 96 x 24
- 132 x 24
- 160 x 24
- 80 x 24 using Hi-Res characters (with little serifs, like an IBM)
- 80 x 32
- 80 x 48
- 128 x 32

Now to my mind that is pretty amazing. In 160 x 24 mode, you get a maximum of 3,840 characters on the screen, in 128 x 32 you get 4,096. That's 4K onscreen.

So how does it look? Surprisingly good. I have a Kaga Denshi green-screen video monitor, of the type very popular with Apples over the past few years, and it was quite readable in all modes. The only problem was a shimmering in those modes which use the Apple high resolution screen, ie the last four of the modes described above.

The excellent manual explains this problem. Because of something called "interlacing", those modes that use the high-res screen need a monitor with high-persistence phosphor, such as the Apple Monitor ///. In the past I have always found this monitor a little disconcerting to use, because the characters stay on the screen for a moment after they are

supposed to have disappeared.

There are two possible problems with certain monitors when using the UltraTerm. One is this shimmering, the other is that of "overscan", of not getting all the characters on the screen when in 160 column mode. Curiously, the manual says that the Apple monitor /// is guilty of this, though it is still the monitor they recommend. My own Kaga was quite legible, if a little small in 160 columns, so I can only surmise that it is a better monitor than the more expensive Apple monitor /// in this regard.

So what can you actually do with all this extra screen space? The characters are very small, there's no doubt about that. You would get very small eyes looking at some of the more compressed modes for long. Eighty columns is now an industry standard, and it's unlikely that you'll ever want to go far beyond that. But you might for some purposes.

One big use I can see for this card is with Visicalc. It is often a big advantage to view a lot of your worksheet, so you can easily see the results of calculations. The makers realise this and you can purchase a separate Visicalc pre-boot disk which uses the 128 x 32 mode, giving you twice as much onscreen as with any other Apple spreadsheet program, including those which use a conventional 80 column card. The extra columns may also be useful for some word processing applications, such as in some legal work, when your page width exceeds the conventional maximum of 80 columns.

I am as unqualified in my praise of this card as I am for the APPLI-CARD and the EPS keyboard.

The EPS Keyboard

There are a lot of us who get frustrated with the standard Apple keyboard because our parents were stupid enough to see that we were born minus the essential three arms and hands with seven fingers and two thumbs on each.

This can be a distinct disadvantage when you want to perform some little trick like CTRL Z Shift C CTRL I Shift and CTRL T which is, please believe us, the series that you go through when you are transferring data through Harry Harper's Vision 80 card, than which there is no better. Several solutions have been suggested and attempted – with varying degrees of success.

Now, from CompuMusic (they of the AlphaSyntauri) comes a professional keyboard which is better than that on the IBM PC.

It has 12 special function keys and a dedicated numeric pad plus four arrows for zapping the cursor around. But more than this, it has a set of EPS Promware models (that's a complicated trade name for an encased EMROM) which, when plugged in, transforms the keyboard into precisely the right configuration for a given program. For example you can plug in Visi-Calc, Applewriter II, Screenwriter II, Wordstar and several others. Once you have plugged in your Prom you have a command template, a sort of plastic overlay, which tells you what each key represents.

This small genius of an idea plugs directly into your motherboard and if you like you can have both your original keyboard and your EPS keyboard working at the same time just in case you want to play duets.

Are there any problems? None. There is a decision you have to make

as to where you will place your computer. But that is not a problem. It's a decision.

We wish that Apple had the intelligence to provide such a keyboard. As they didn't, Executive Peripheral Systems did and CompuMusic brought it to Australia.

To sum up our attitude: we are buying one. With money.

The 88-CARD

Another card imported by CompuMusic which we have not had a chance to test is the 88-CARD, also from PCPI. This card contains the 16 bit Intel 8088 microchip and allows you to run MS-DOS and CP/M-86 16 bit operating systems. While you can't put an IBM PC disk straight into your Apple, you can download software from that machine and run it on your Apple. This obviously opens a whole new world to the Apple II, and with some extra memory and a larger disk drive (and a better keyboard, etc) the Apple II is in every way the equivalent of an IBM PC.

We look forward to the opportunity of testing this card, which we will as soon as they become available. In the meantime we are left to ponder just what this new generation of peripheral cards means for the Apple II.

Where's it all going?

There are over one million Apple II and Apple //e computers in the world. That's a lot of computers in anybody's language. A lot of people who have Apple II's are aware of the advances in computing technology since that machine was designed, but they are also aware that in the Apple II they have the best small computer ever designed. Many of them will be so attached to their Apples that they will never get rid of them, no matter what.

If you're one of these people, as am I, who want to keep their Apple while not missing out on the advances that are happening in computer technology, it seems that the types of cards that we are reviewing here are the answer. It is fortunate indeed for Apple that the massive number of Apple IIs in the world today means that there will always be designers who will come up with new cards and peripherals to keep the machine abreast of technology. It matters not that the Apple uses a different keyboard, different RAM, a different microprocessor, different disk drives, different whatever. Somehow it's still an Apple. □

Numeric keypad

by Bevan King

Have you ever wanted a numeric keypad for your Apple just because other computers have got it? Well, I did. But since my budget is a bit tight, I abandoned the thought of buying one. One Day, as I was looking at Tandy's catalogue, I came across their new portable computer. It has a numeric keypad on it (as usual), but since it's a portable it uses the "M", "J", "K", "L", "U", "I", "O", "7", "8", "9" to form the numeric keypad. So off I go taking out my assembler, my Apple II reference manual, and finding information about the part of my Apple that I always thought was too hard to understand.

This is version 2.4 because the previous version didn't work as I expected. The pointer in zero page $06 and $05 is just to check if the computer is in "KEYPAD" mode and the number of ctrl-s's to be removed from the buffer. The explanation in the source code is quite straightforward. To activate it, first BLOAD KEYPAD2.4, then do a CALL 779 to redirect the keyboard input switch, and from now on, a ctrl-s will turn the keyboard into a numeric keypad using the "M", "J", "K", "L", "U", "I", "O" keys to represent "0", "1", "2", "3", "4", "5", "6" respectively. In addition, the "+" and "*" keys don't have to be shifted. To quit this routine for good, type CALL 768 or simply hit reset.

This routine is relocatable, and can be executed either in immediate or deferred mode. If you relocate this routine, then of course, the location of the CALL would have to be changed.

For those of you who are using Program Line Editor in your program, after you do a CALL 779, all the special functions will not work. The only way to get out of it is to hit RESET. (Well, if you don't like the "beep", just do CALL 768 followed by a CALL PEEK(1010)+PEEK(1011)*256. A bit of bad luck there, but I'm working on that problem now. Just hope there aren't that many people who'd use PLE in conjunction with this program!

I don't do a lot of data inputting but occasionally I need help to type in a lot of numbers when writing a program eg 1000 DATA 10,20,30, 2,3,55,67,22,77,-255,564,55 ... etc. With the help of the keypad, this is much more convenient and reduces frustration time.

Experienced programmers may like to change this program so that using the "Q", "W", "E", "A", "S", "B", "Z" would give a push-button telephone keypad.

I hope this program helps when inputting numerical data. □

The Datalife Disk Analyzer

Verbatim are well-known for their disks, especially for Datalife, but not many people know they produce good software as well.

The Disk Drive Analyzer is a very high quality program written to check the performance of your disk drive. It runs on all Apple II series and Apple /// disk drives, and all Apple compatible drives. (It requires 48K RAM, but nowadays, who's interested in how many "K"s it requires?)

All you have to do is boot the disk up and all is ready to go. The disk has to be left in the drive all the time, otherwise, a message will show up requesting this. One good thing about this program is that if you leave the drive door open, the drive won't give you that heart-breaking noise; it quietly stops the motor and asks you to put in the Disk Drive Analyzer disk.

Upon booting up, a title comes up with "Datalife" moving from right to left and after a while the menu page appears. It gives you six options.

Option 0: this is just an introduction explaining what each option does.

Option 1: Radial Alignment Test. For those of you who don't know what this is, it simply means that the head of the drive has to be centred over the correct track location to read and write your data accurately. The screen will show a side view of the drive, and you can see the head and disk. When the test starts, the head is positioned around track 17, moves inwards to track 34, back to track 17, then moves outwards to track 0. The result is then displayed on the screen, giving a rating of good, fair or poor.

Option 2: Disk Speed Test. This is the most common test seen on other programs like Locksmith, Nibbles Away. But the one on the Disk Drive Analyzer is different from all the others because it doesn't show you what your drive speed is,

it just shows an Apple drive with a disk spinning in it. That's it. When the test is finished, it also tells you whether the speed is good, fair or poor.

Option 3: Disk Clamping Test. The clamps are the two cups you see when you look into your disk drive. This test checks if the mechanism clamps the centre hole properly. On the screen you see the top of the "clamp" slowly moving down, then the disk spins, and after a few seconds the drive stops, the clamp moves up, and the result is displayed. Again, it tells you if it's good, fair or poor.

Option 4: Read/Write Test. This is the fastest read/write test I have ever seen. It must be 200% faster than the read/write test of Applecillin. The screen shows a line being passed from the drive head to the

disk, and after a while, the line (information) is transferred back to the drive head. All this taking less than one minute. Your drive will either pass or fail this test.

Option 5: Autotest. This option performs all the above mentioned tests automatically in less than two minutes. Unlike the other test, this one doesn't have hi-res display.

The program is very user-friendly, giving you on screen help and prompting. The hi-res animation is excellent, giving you some ideas on what your drive is doing.

Professional technicians may not find the Datalife Disk Drive Analyzer useful, but for people like me who are always worrying that their drive is "going south", this is a very good utility program. Once you see this program, you'll want to buy it. □

No computer has ever been as efficient as this.

In many ways, your office desk still remains the most productive way to work.

Everything's at your fingertips. The whole job's laid out in front of you. And you can do 10 things at once.

Yet, every computer ever devised has ignored this. Forcing you to work the way <u>it</u> wants.

(Assuming you can plough through the manual and learn a complex language of computer commands.)

New Apple Lisa is the very first personal computer designed to operate the way you do.

All of its awesome power isn't devoted to making electrons go faster, but to helping <u>you</u> go faster.

By working <u>visually</u>.

Complex computer commands are replaced with familiar pictures, or symbols.

Files look just like file folders, memos like memos.

There's a calculator, a clock, a clipboard – even a wastebin – all in symbol form.

So, even if you've never gone near a computer before, instead of taking at least 30 hours to learn (as conventional personal computers do), Lisa takes less than 30 <u>minutes.</u>

To tell Lisa what you want you just point to the appropriate symbol, using a palm-sized device called a "mouse".
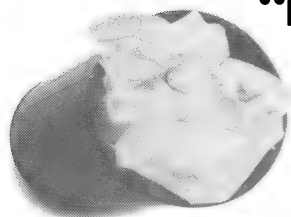
Move the mouse across your desk and the pointer moves across the screen.

No longer are you totally dependent on the data processing department.

Effortlessly you can turn figures into graphs, draw diagrams, paste worksheets into reports. And print out the lot while you get on with something else.
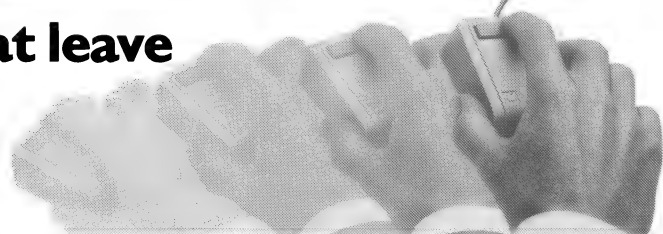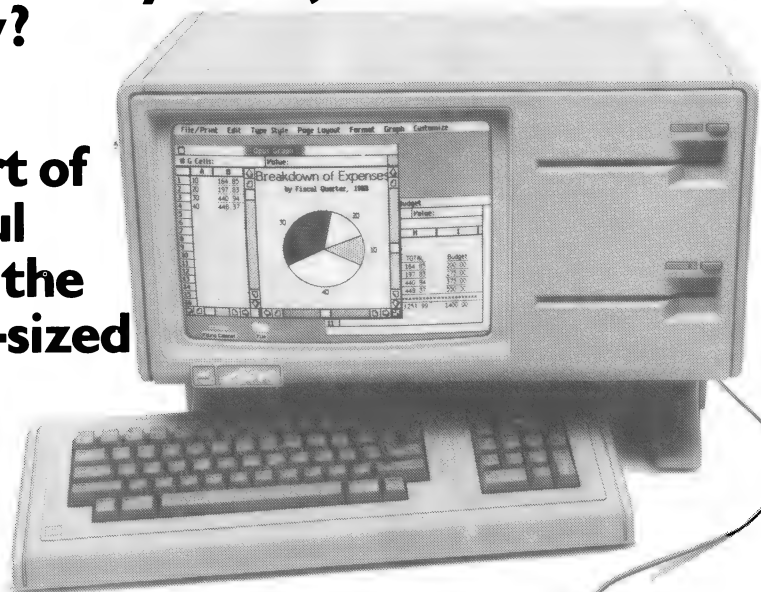
It's what computers should've been all along.

Does it all sound too easy? Is it just some kind of high-priced executive toy?

By no means.

Lisa is nothing short of the world's most powerful personal computer, with the internal memory of a mid-sized mainframe.

You may even feel Lisa could replace your desk entirely.

Except what would that leave you to put your feet up on?

Rounding in Basic

by Graeme Philipson

BASIC is in many ways a great computer language. It is very easy to learn, in fact it was originally designed as a language to teach people how to program. The letters "B.A.S.I.C." stand for "Beginners All-purpose Symbolic Instruction Code". It is also suited to a wide range of applications: just about any programming job can be written in BASIC. Many other languages are much better suited for particular jobs, but few match BASIC's flexibility. It is not surprising that it has become the most common language for microcomputers (and therefore for computers in general, because there are far more micros than anything else).

But BASIC has many disadvantages. It is comparatively cumbersome, and it is "unstructured": ie it lacks any internal order. Ask a dozen people to write a program in BASIC, and they'll write it a dozen different ways. It makes it very hard to understand other people's programs.

Whatever BASIC's pros and cons, we are stuck with it. It is the first language learnt by virtually all Apple owners, if only because it is built into the machine. Once the novice programmer gets beyond "PRINT" statements and learning how to perform simple calculations, problems start occurring, many of them because of the language's inherent weaknesses. A very common such problem is that of number rounding.

Round and round she doesn't go

When calculations are performed on numbers in BASIC, the results are calculated to as many decimal places as necessary, up to 8. For example, if you divide 4.5 by 3.7 you get the answer 1.21621622. Both divisor and quotient have one decimal place. but your answer has 8.

This is often not a problem. For mathematical calculations you want all the accuracy you can get. But it can be a little untidy when calculating monetary amounts, particularly if you want to print your results in nice columns, every amount with two numbers after the decimal point. That can be tricky but it is still not that difficult.

Rounding to integers

Rounding numbers so that they don't contain too many decimal points is very easy. Applesoft BASIC, like most BASICs, contains an INT function which rounds numbers down to the number below. For example

INT 5.46
returns the value "5". The trouble is,

INT 5.999

also returns the value "5". What you need to do is add .5 to each number before you find its integral value. This will then return the integer CLOSEST TO the original number, rather than the integer BELOW the original number. The form of the expression is

X = INT (Y + .5)

where "X" is the rounded number and "Y" is the number you want rounded.

So far so good. This is still basic BASIC. Things become a little more difficult when you want to round numbers to a certain number of decimal places, for example two, or six. In this case we have to do a little juggling, by adding a smaller figure and multiplying and dividing. To get a number to round to two decimal places, we use the expression

X = INT ((Y + .005) * 100) / 100

Similarly, to round to three decimal places use the expression

X = INT ((Y + .0005) * 1000) / 1000

Unwanted fractions

Sometimes when performing calculations in Applesoft BASIC you will get small fractions on the ends of numbers which should not be there. A calculation to which you know the answer to be "5" might return the value "5.000001" or "4.99999999". This is because of the method computers use to calculate: they use binary numbers, we use decimal. When a computer converts two numbers to binary, then performs calculations on them, then converts them back to decimal, it is possible that small discrepancies may occur through the compounding of almost infinitesimally small errors brought about by inexact binary-decimal conversions. Using the above rounding procedures will get rid of this problem.

The Apple as a computer terminal: Communications packages for the Apple II

by Graeme Philipson

An increasingly popular use of microcomputers, the Apple of course included, is as terminals. With the addition of a few bits of hardware and software the Apple II can become a complete intelligent terminal, capable of communicating via normal telephone lines with mainframe computers, databases, electronic mail services – anything that you can use a conventional terminal for.

What sorts of things? Many people are still unaware, or only vaguely aware, of what a computer terminal can do. Despite being touted for some time, with such phrases as the "Global Village", the "Electronic office", "Paperless communication", the field of microcomputer telecommunications is still in its infancy, mainly because of its relatively high price compared to more orthodox means of communication.

But this is changing very quickly as telephone charges drop and microcomputers become more widespread and less expensive. As an example, a lot of publicity has been given recently to unauthorised entry of databases in the US by teenage "hackers", as in the recent movie "Wargames". The time is coming when inter-computer communication will be as common as a telephone call, though that day is further away than a lot of techno-gurus seem to think.

Let's have a look at what you can do NOW, with an Apple II and less than $1,000 worth of software and peripherals.

1. You can access any of the public database/electronic mail services. There are two well-known ones, one American and one local. "The Source", owned by Reader's Digest and based in Virginia in the USA, allows anybody to access any of the data available on their large mainframe computer. There is a vast amount of information: weather reports, news, computer problems. You can also leave messages for other users by accessing their number. They then collect the messages when they connect themselves. You can scan a list of users currently using the system, and break in and "chat" with them if you, and they, so desire. And there is a lot of advertising, omnipresent in today's world.

There is an Australian attempt at emulating The Source's capabilities, called "The Australian Beginning" (TAB, they call themselves). This is Melbourne-based, and has suffered a few problems in its short lifetime. It is quite good, though it is very limited compared to The Source. Its electronic mail facility is cheaper, but it has never caught on with the business community.

2. You can access any of the extremely large number of semi-public databases, most of which are American-based. There are a few available in this country, but once again their use is limited by the relatively scarce amount of information available. These databases are considerably more expensive to access than The Source or TAB, but they are extremely useful for a number of specialist functions. A good example is the "World Reporter" recently introduced in Australia by Software Sciences. This database allows a user to access news items that appear in the British Press. Just type in a codeword, say "Lebanon", and the location of all news stories which use that word will be displayed.

Overseas databases are accessed via OTC's "Midas" network, a communications link which is much cheaper than having to make an international phone call. But the information still has to go via satellite, and it is prohibitively expensive for most private users.

3. You can access your company's or your university's mainframe computer. This allows people such as computer programmers to work at home or off-site. In this situation the microcomputer usually acts as a "dumb" terminal, performing the same role as a terminal attached directly to the computer. The only difference is that communication is over a phone line, rather than directly into the computer.

4. You can interact with other microcomputers. If you and a friend both have a terminal, you can ring up, say something like "I want to send you a file", you both put your computers into communications mode, and you send your text file through the phone lines directly into your friend's computer. We use this method ourselves to collect some of the stories for this magazine.
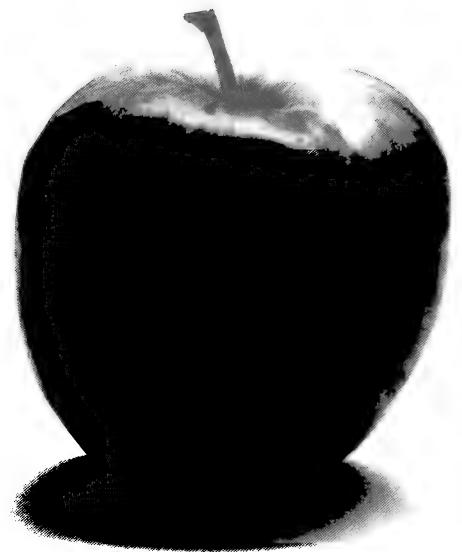
What do you need?

You need hardware and you need software. Sound familiar? It's the old story of the hardware being no good without the software to support it. The hardware consists of two items:

1. A serial interface card. Terminal communication usually occurs via what is delightfully called an "Asynchronous RS-232C Serial Interface". In the Apple II's case, this consists of a plug-in interface card which normally likes to reside in Slot 2. There are a number of such cards available: the one we recommend is the "Digitek" serial interface card. It costs around $200, which is a lot cheaper than some, including Apple's so-called "Super Serial Card", and it is well-made and easy to use. It is commonly available, and any Apple dealer should be able to supply one.

2. An acoustic coupler or a modem. An acoustic coupler is a device which plugs into the serial interface at one end and a telephone at the other. You have to mount the telephone handset in the little rubber pockets, which can be a bit cumbersome to do, even impossible if you have a non-conventional handset. The acoustic coupler converts the electrical impulses into high-speed clicks which are transmitted to the telephone's microphone, where they are once more converted to electrical signals. The same procedure happens in reverse at the other end, or when you are receiving data. It is inherently inefficient, but at low speeds (typically 300 Baud, ie 30 characters a second) it works okay. The big advantage of acoustic couplers is their portability. As long as you have a telephone, you have a terminal connection.

The alternative to an acoustic coupler is a modem, which stands for MOdulator-DEModulator. This performs the same function, but there is no conversion from electrical to audio and back again. You need to plug your modem into a telephone

outlet, coupled with a telephone. This requires a special plug which can only legally be installed by Telecom, which reduces the portability of modems. It is possible to do the wiring yourself, but this is highly illegal and still not portable. As this is a respectable publication, we cannot recommend this course other than to say that we know of people who have done it themselves and it does work.

Modems are more efficient, which means they can run at much higher speeds. This can mean a substantial saving, depending on what proportion of your log-on time is spent actually transmitting and receiving data.

Prices of acoustic couplers and modems have fallen recently and range between $200 and $500. The best acoustic coupler has long been the "Sendata 700" by Anderson Digital, at $249. They also make modems, the cheapest of which is the "Sendata 300", for $220. Dick Smith (The Electronic Dick!) has a good little modem for $199.

The software

The hardware, as is often the case, is no real problem. We haven't gone into it too much, because hardware is hardware and if you purchase a well-known brand from a reputable dealer you can't really go wrong. But software . . . that's a different story. There are many different programs, called "terminal emulators", available. They vary in price and capability, and which one you get will probably depend on which one the salesman wants to show you. We look at a few of them later in this article: the list is by no means exhaustive, but it includes the most popular packages plus a few new ones, so it should give you some indication as to what to look for in a terminal emulator.

One very important piece of advice: before you buy, see the software you want actually running with the hardware you have connected to the "other end" you will be using. Make it a condition of the sale that you see it working, preferably on your own machine. There can be fiddly cable connections, depending on your modem and serial card, there can be tricky bits to actually getting the

software running properly so that your data is transmitted and received clearly, there is usually a lot that needs to be done. This is because you will usually be using different bits of hardware and software from different manufacturers which should ideally all be compatible, but which can require a bit of tweaking to get working properly.

Don't be daunted by this however. Once it's working, it should be trouble-free. If your dealer can't or won't help you, find one that can. That's what computer dealers are for.

What to look for in a terminal emulation program

Any Apple II communications program should do the following things:

1. Allow you to transmit Apple DOS text files. This allows you to create documents offline using a conventional word processing program (eg Zardax), then send them at high speed once connected. This means that you do not have to key your document in when you are logged on, which has obvious advantages of time and therefore money.

2. Allow you to use a "Log-on File". When you access a data base, particularly if you go via Midas, there are many access numbers and code words which you have to type in when you connect. If you have to do this often it is a time-consuming, boring and repetitive task. A log-on file allows you to store all this information in a text file, where it can be retrieved when necessary and do all the boring work for you.

3. Allow you to save your terminal conversation as an Apple DOS text file, thus allowing you to edit or print the document once you have logged off.

4. Have facilities to use different types of data transmission. There are a few different standards of serial data transmission which need not concern us here. They have to do with such things as different speeds, different check characters, different duplex settings.

5. Operate in 80 columns. While this is not necessary, most professional quality databases use an 80 column standard for the formatting of data. If your terminal operates only in 40 columns some data may be less legible. □

LOGO – the cry of the Turtle is heard in the land

by Ian Gronowski

In recent years Logo has become one of the most popular teaching languages. Designed at MIT by a team headed by Seymour Pappert, it has been successfully implemented on most micros, though the versions for the Apple have received the most acclaim.

The hardest thing about writing an article on Logo is to try to prevent it from turning into a review of "Mindstorms". "Mindstorms" is a book by Seymour Pappert in which he sets forth his views on education. One paragraph particularly bears quoting:

"In many schools today, the phrase "Computer-Aided Instruction" means making the computer teach the child. One might say the computer is being used to program the child. In my vision, the child programs the computer and, in so doing, both acquires a sense of mastery over a piece of the most modern and powerful technology and establishes an intimate contact with some of the deepest ideas from science, from mathematics, and from the art of intellectual model building."

Logo provides the learning environment, the context with which the child learns how to reason. The idea of programming is introduced through the metaphor of teaching the computer a new word; and with this model the child develops naturally the idea of structured programming, of thinking procedurally. General problem solving methods – heuristic strategies – are encouraged by playing with Logo.

Also rather than using the emotionally loaded terms of "right" and "wrong" and consequent shame about failing, using a computer results in the neutral concept of the "bug". Errors become a necessary by-product of learning, not a child's "mistake". Most importantly, Logo is fun.

The Turtle

The central theme of Logo is to provide easy entry into the world of programming for non-mathematical and hence pre-mathematical children. It amounts to a philosophy of education. Ease of entry is provided by the "Turtle". A Turtle is an object with several characteristics, such as position and heading. It is like a person or an animal. Children can identify with the Turtle, especially as the Turtle is usually either an actual physical robot with wheels, a dome shape and a pen for drawing, or a "light turtle", a triangular object on the screen. These two different physical objects are mathematically the same or "isomorphic".

The actual commands for controlling the Turtle are FORWARD and BACK, for movement, and RIGHT and LEFT to change the Turtle's direction – to pivot it. Thus a square with side of length 100 units could be drawn by typing

```
FORWARD 100
RIGHT 90
FORWARD 100
RIGHT 90
FORWARD 100
RIGHT 90
FORWARD 100
```

Procedures can be defined, ie, we can teach the Turtle new words:

```
TO SQUARE : SIZE
    REPEAT 4 [FORWARD    SIZE
RIGHT 90]
END
```

This means "to draw a square, go forward the size of the side, then turn one right angle, and repeat this four times".

Now, just as we had previously typed in FORWARD 100 we can type SQUARE 100, to draw a square, as if it too was a "primitive procedure". We have no way of telling that SQUARE, and not FORWARD was designed by the user. (Not strictly true). From simple beginnings we already have the powerful concept of modularity, of creating small conceptually distinct parts of programs that can be reused.

When considering Logo, it is important to emphasise that it is designed as an educational tool, and not something to write databases with. This leads us to consider different issues than those considered when learning, say, FORTRAN.

Turtle geometry is "intrinsic". It only knows the local area, where it is

at. This is opposed to the usual, global nature of school mathematics. Whereas in Cartesian geometry we define a circle as the locus of all points a given distance from a fixed point, the centre. It is represented by the equation:

$$(x-a)*(x-a) + (y-b)*(y-b) = r*r$$

With Turtle geometry a circle is defined in terms of constant curvature (curvature is how much we turn for each little bit we move):

```
TO CIRCLE
    REPEAT [FORWARD 1 RIGHT 1]
END
```

Thus, to draw a circle take a little step forward and turn a little, and keep on doing it. This is more intuitive, and more powerful than Cartesian geometry. In terms of computer power it is also more useful. A Turtle Geometry definition is "imperative" – it defines a circle, and it tells the computer how to draw it. A Cartesian Geometry definition is "declarative" – it describes a property of a circle. In order to actually draw a circle from a Cartesian definition, you must translate that definition into a more useful form. The equation of a circle does not tell you how to draw it.

The Turtle is an intuitive explanation of the heart of the ideas that lie at the foundation of the calculus, of differential geometry, and of modern physics. The Turtle is the analogue of the differential equation. With a Turtle

we can easily simulate both Newtonian and Einsteinian physics.

LISP

We leave the Turtle for the moment to consider the other source from which Logo springs. This is LISP (LISt Processing language), and the core of Logo. Its syntax and style are derived from the granddaddy of all Artificial Intelligence languages.

In LISP the fundamental form is the list, an ordered sequence of arbitrary LISP objects, ie words or other lists. Thus, examples of lists include [3 2], [[5 7] [3]]. and [FORWARD 100]. (In LISP, parentheses are used to delimit lists, in Logo brackets.) The distinction between procedures and data, tight and rigid in languages like Pascal, is blurred completely in LISP and Logo. Procedures themselves are stored as lists, as in the last example. They can be manipulated like any other data: modified, passed as parameters to other procedures, and returned as results. So, we can easily write a procedure, say CREATE-N-GON which creates a procedure which draws an n-gon. Thus (CREATE-N-GON 4) would result in a procedure 4-GON, which draws squares when called; (CREATE-N-GON 12) one which draws dodecagons, and (CREATE-N-GON 200) one which draws (near) circles.

The other feature drawn from LISP is recursion. The classic, if trivial example of a recursive algorithm is the definition of "factorial n", written n!.

$$n!=n*(n-1)*(n-2)* \ldots *4*3*2*11, \text{ if } n$$
greater than 0; 1 if n=0

The recursive definition of n! is derived by noting that if n greater than 0, n!=n*(n-1)! and if n=0, n!1. In Logo this becomes

```
TO FACTORIAL :N
    IF :N = 0 OUTPUT 1
    OUTPUT :N * FACTORIAL (:N-1)
END
```

So far, the only Logo objects explicitly described are numbers, lists, and procedures. Two more fundamental objects are characters and words. Words are lists of characters,

characters are standard ASCII. The list primitive operations such as FIRST, which returns the first element of a list, and BUTFIRST which returns the list consisting of all but the first element of its argument can be use on words and characters as well. Hence

FIRST BUTFIRST "LOGO

returns an 0.

To quote Douglas Hofstadter, "Lisp is Crisp". Unfortunately, I can't come up with a nice rhyming three word summary of Logo. So, bearing in mind the richness and power Logo inherits from LISP, its brilliance as an educational language, a vehicle for the easy digestion of deep ideas in byte-size chunks, we might sum up with this (recursive) acronym for LOGO:

LOGO Offers Great Opportunities

Bibliography:

"Mindstorms", Seymour Pappert.
This book is a must for all those interested in Logo, Computers and Education, or theories of learning. Well written, a joy to read.
"How to Solve It", George Polya.
Another book which all aspiring computer programmers, mathematicians and educators should read. A classic.
"Goedel, Escher, Bach", Douglas Hofstadter
Monstrous clever linking of the music of Bach, the etchings of Escher, the metamathematics of Goedel. The book consists of text interspersed with dialogue, the dialogue between Achilles and the Tortoise illustrating the ideas with witticisms, nested puns, self referential jokes, and recursive acronyms. A beautiful book.
"Byte", August 1982
August's "Byte" is the language issue. In 1982 it was devoted to Logo.
"Turtle Geometry", Abelson and diSessa.
An essay mathematics text, using Turtle Geometry and Logo to teach vector calculus, relativity, and other fun Turtle stuff. High School level (an average HSC student should have no difficulty with it).   □

As some of our more eagle eyed readers will know, the publisher of this magazine also writes regularly for the "Weekend Australian" on microcomputers. Some weeks ago he wrote an article on the advantages of the Dvorak keyboard over the QWERTY keyboard. To his chagrin, he now finds that the "Weekend Australian" is not, as he thought, compulsory reading for every Apple owner, and for those who missed it here, as they say, are the main points of the news.

Those who have read it please duck under the main article and have a look at the letters at the end.

Qwerty/Dvorak

In 1873 Christopher L. Sholes invented the typewriter. His invention looked like a sewing machine with typebars striking upwards to the roller. This meant that the typist couldn't see what was being typed which led to all sorts of problems.

But the biggest problem was that the bars operating the keys lay close together and if a typist hit keys operating adjacent bars too quickly, the two bars stuck together through friction. Those of us who started with steam typewriters will remember that this fault remained with manual typewriters to the bitter end.

To get over this problem Sholes, an exceedingly ingenious inventor, studied the frequency with which letters were used in the English language, and then designed a keyboard which slowed down the fastest operators so that they couldn't jam the keys.

This keyboard, which most typists will recognise as QWERTY, became the industry standard — and is still the standard used on modern computers, which is a situation approaching lunacy. Computers do not have key bars that jam together through friction.

Very shortly after Sholes' machine hit the market, several inventors came up with better keyboards which improved the speed of the typist. But Sholes' splendidly designed piece of inefficiency won the day. He was, after all, there first.

In a book called "The Wonderful Writing Machine", Bruce Bliven writes: "From the standpoint of the touch typist, this arrangement of the alphabet is madly inconvenient. According to the many engineers, psychologists and student PhDs who have studied it, the

standard keyboard is considerably less efficient than if the arrangement had been left to simple chance."

Quite so. That is the way Mr Sholes meant it to be.

Just before WWII, Dr August Dvorak, then professor of English at Washington State University in the US, decided to do something about it and got a navy contract to improve the typewriter keyboard. Why the navy I know not, but an improved keyboard was ready for testing in the early '40s.

The Dvorak keyboard was found to be 20 times easier to use than the Sholes. Typically a learner can be an extremely efficient touch typist in a quarter to half the time it takes to become proficient on an average keyboard. And most users of Dvorak keyboards average 100 words a minute with ease. Typists on normal keyboards are considered good at half that speed.

And did the world jump on the bandwagon and convert all typewriters to Dvorak keyboards, thus saving untold millions of dollars and work hours? They did not.

There are two reasons for this. Secretaries who learned to touch type on the QWERTY keyboard were jealous of their skills. They saw no reason to re-learn a new method, even ·if it was demonstrably twice as good.

So although the Dvorak keyboard was commercially available in 1944 — more than 10 years ahead of the first serious computer — typewriters remained unchanged. And this keyboard became the standard inputting system — because in the main programs are typed into machines and typists with QWERTY experience were the ones that got the jobs.

That doesn't mean to say that the

Dvorak system completely disappeared. All the world typing records were made using Dvorak keyboards, IBM and Smith-Corona offered typewriters in the US (but not, as far as I can ascertain, in Australia) with Dvorak keyboards.

The situation at the moment is that no computer manufacturer that I have been able to find is offering an improved keyboard.

All the children in Australian schools who are being taught the basic skills of computing, are being taught on a keyboard which was designed specifically to slow them down. Not a happy thought.

How difficult is it for a keyboard to be given a Dvorak conversion?

Typewriters I know little about, but with most microcomputers it is possible to write a software program that changes the keyboard from the inefficient Sholes to the efficient Dvorak.

Felix Macri has written one for Apple users. The vast majority of microcomputers have keys which are a press fit and it is easy to lever them off and change them around to a Dvorak pattern.

What, of course, is really needed is for the computer manufacturers to throw off the shackles of the 19th century, realise that they are using a keyboard which was designed to be inefficient in 1873, and issue their machines with Dvorak keyboards. The vast majority of first-time computer buyers are not touch typists, so nothing is lost.

Indeed, it is worth noting that touch typists who are fluent on the QWERTY board have no difficulty in switching backwards and forwards between the two systems. It is rather like driving a car with a stick shift gear box and then switching to one with an automatic gear box. The mind readjusts very quickly to the movements that are needed.

This subject is plainly of interest because there were 868 letters written to the "Weekend Australian" immediately after the article was published. Some points raised in those letters were interesting.
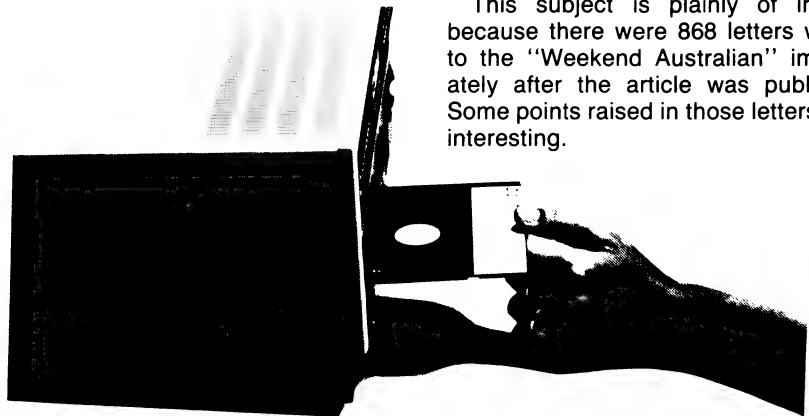
Dear Sir,

I was interested in the article on QWERTY versus DVORAK. I use my Apple mostly as a word processor (with Zardax), and it certainly would be useful to be able to type faster. But I was trained as a touch typist on the QWERTY keyboard, and my speed is good already. It may be that I would become confused between the two keyboard layouts and thus casue the speed I already have to drop.

I think it would be unlikely that I could swap between the two methods. But please send me a program, I'm going to hae a try anyway.

E.L. Jones

Dear Mr Powell,

Some time ago as an accountant (now retired) I did some research into improving keyboards to coincide with the new electronic machines. To me it seemed ludicrous to have such a layout as the DVORAK and not take advantage of it.

I discovered it is not the inability to produce such a keybard that prevented its introduction; typewriter manufacturers can readily produce any keyboard your heart desires. Brother at their Pitt Street store had a display last year of typewriters with keyboards in Russian, Arabic and various special combinations – if demand is there it can be cone fairly easily. However the electronic age introduced the versatility of an alternative keyboard in each machine – witness your offer of a suitable software for adaption – this of course cannot be done with a mechanical machine.

My idea was to introduce all electronic wordprocessors and such like with the built in facility of an alternative keyboard so that by means of a switch and suitable overlay any operator trained in whatever mould could use the machine.

Your comments about changing from one style to another is interesting because my research indicated that reluctance on the part of operators trained on the QWERTY system to try another layout was the basic reason for non-introduction of the more efficient system.

In my opinion the only way to effect a change is for official encouragement at say Government level, others would then follow. With alternative keyboards old diehards could still use the old system until they retired.

You may be aware that in the USA the Oregon State government stipulates in its tender requirements for keyboard contracts that DVORAK layouts be supplied. ("Electronic Office" Australian Financial Review 27/7/81).

It is articles such as yours that may eventually sway public opinion and ultimately business houses so that the system can be improved.

P Scully

Dear Mr Powell,

Have you read Seymour Papert's comments on QWERTY? Enclosed is a snippet quoted from a scientific discussion written by a colleague and myself:

If, in view of these facts and the history of the origin of Mancy's definition, most workers in the oxygen sensor field should be satisfied with a "Mancy" definition" of permeability, there is the danger of receiving the condemnation of the following words of Seymour Papert (Mindstorms: children, computers and powerful ideas, Harvester 1981, p 51) in another context:

"But, like QWERTY, it has dug itself in so well that people take it for granted and invent rationalisations for it long after the demise of the historical conditions that made sense of it."

(Of course, QWERTY is only unnecessary, and a damn nuisance, not misleading.)

David Short

Dear Sir,

I give my best wishes to the success of your "local" Apple magazine.

At present it is young and slender but I have every confidence that as it matures it will put on weight (much as we all seem to do). Certainly the C. Itoh 8510 Printer Utility in December's edition was, to me, in itself worth far more than the $3.00 cost of the magazine.

I for one much prefer to see a magazine of this specialist type with useful and interesting information without undue "waffle" and padding.

Now my query. I have noted readers' letters regarding Zardax and would be most interested if anyone knew whether, and if so how, I could incorporate the "Diversi-Dos" print Buffer Utility into Zardax. I have an Apple IIe and a C. Itoh 8510 printer. If such a software controlled print buffer were able to be married into Zardax that program would be an unstoppable world beater.

M. Bennett

Ed. *We don't think this is possible, but if anyone knows how to do it please let us know.*

Dear Sir,

I was very pleased to see an Australian magazine written for Apples (and presumable compatibles). I was, however, disappointed in your article about "fake" Apples. Don't get me wrong, it was very interesting and extremely informative but it told a little white lie! It stated ". . . Medfly. More on this later." I searched and searched but to no avail! Ah-ha, I said, they mean more on this later in the next part of this article in the next issue. I have just bought the second issue (which appears to have put on a little weight, too much good food at Christmas time) and read the follow-up, STILL no mention of the Medfly.

I await with baited breath your answer to this letter. I am very pleased with my machine but I would be most interested to hear of your and others' experience with it.

Andrew Merrylees.

Ed. *We are* **definitely** *reviewing the Medfly in the next issue.*

The following story is not, I am assured, true. The promotional film for the new Macintosh shows a scene from, I think, George Orwell's "1984".

There is a theatre full of grey zombies being harangued by a Big Brother on the screen who bears a passing resemblance to a surly Nikita Krushchev. All this is in the full glory of black and white.

Then into the auditorium bursts Wonder Woman (at least she looks like Wonder Woman to this Worm's tired eyes) carrying a sledge hammer. She races to the screen and smashes it to reveal, in its full glory, the Macintosh.

This is all obviously allegorical and well over a poor Worm's head. But the story I heard from my American Cousin through the "Worm Source" was that in a cost cutting exercise in Cupertino they decided to use the senior executive staff to play the part of the grey zombies being harangued by Big Brother.

This I find difficult to believe. Even more difficult to believe is that none of them needed to wear make-up of any kind or change their clothes and that their mumbling voices were taken from a tape recording of an Apple board meeting.

Can this be true, I ask myself? Certainly not.
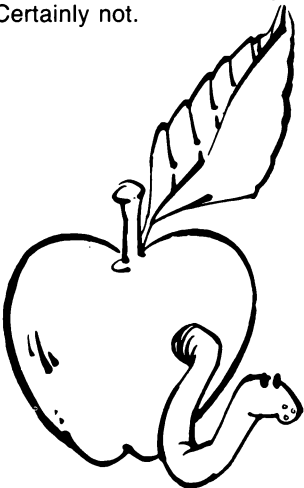
Press releases

When the Macintosh was launched sundry press releases were put out by Burson Marsteller, a company dedicated to destroying the English language within the span of this century. And they are succeeding.

They issued an Apple Macintosh "Backgrounder".

That word is not in the Concise Oxford Dictionary, nor yet the Shorter Oxford Dictionary. Not even in the full 24 volume edition. Look for it not in Websters. Fowler finds it a mystery. Sir Ernest Gowers knows it not.

Can it be that Burson Marsteller are inventing new words? Then they should stop. "Backgrounder" does nothing to enrich the language of Milton and Shakespeare.

And they are not overly strong on figures. In one handout — called "Backgrounder: Apple Computer Australia" — they say "Apple Computer Australia took over marketing and distribution from its former distributor, Eletronic (sic) Concepts, in May 1983. Within the year, the local company claimed sales increases of 00 per cent". Now if that 00 per cent increase that they claim is correct then it is no wonder that Electronic (I'm almost positive that is the right spelling) went bankrupt fairly recently.

In the same handout it says "By 1984, about 1.4 Apple series personal computers were in use world-wide". Well this Worm had one Apple computer. Will the poor wretch who only had .4 of a computer please write in and I'll see what we can do to help.

Strange addition

This Worm accepts that addition is not one of his stronger points. Worms have no toes to count on. But regard the following figures and work them out on your calculator.

A year ago this Worm was assured that there were 35,000 genuine Apples in Australia. Apple management used the term "user base" which was probably specially invented for them by Burson Marsteller.

In the press handouts that accompanied the launch of the Macintosh at the beginning of this year the figure mentioned was 30,000. Which suggests a reverse sales campaign of great magnitude.

Shortly after the launch Apple found themselves in legal strife. They quite rightly, in this Worm's humble opinion, appealed against the judgement which said that Computer Edge could bring in Wombats from Taiwan as legally as Prime Minister Hawke will send Wombats to Japan.

In the Appeal they were represented by the splendid Queen's Counsel, Mr McHugh. He, as a Queens' Counsel is also an Officer of the Court. And as an Officer of the Court he, like George Washington, cannot tell a lie.

And in court he announced that there were 20,000 Apples in Australia. This Worm checked the court records and there it is in black and white.

Who, this pathetic Worm pleads, is one to believe?

If all the figures quoted are correct this must be the most successful marketing campaign in reverse that the microcomputer industry has ever seen. Perhaps Burson Marsteller were correct when they said an increase of "00 per cent". Perhaps I have been wrongly maligning them. Perhaps.
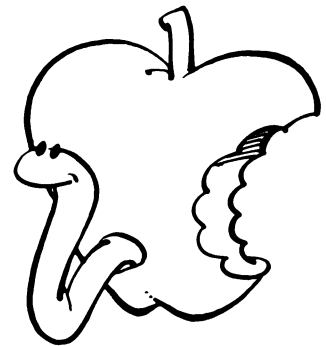
Education cunning

Investigating the educational world of microcomputers, I was interested to see one series which improves the mind by using a combination of intelligence and low rat cutting. I refer to the Plato series which appeals to the avarice in every child by working the questions around the question of money. In their physics lesson they start the pupil off with $25 and deduct $2 for every question the pupil has to ask before the right answer is typed in. Some may feel that this encourages avarice. I do not. I believe it introduces a child at the right age to some of the sordid realities of life — and at the same time teaches physics in a way that will never be forgotten.

What I do object to, in a humble Worm-like way, is another company that teaches modern history by having the pupil emulate the war in Vietnam. I was lucky, Worms rarely get drafted, but I truly shudder when I see children being

told that they must make decisions about "eliminating traitors" and "extending the draft to 17 year olds".

There is no telling where this sort of thing can end. They will be teaching them how to cook earthworms next.

The Worm will turn

It is not widely known that I am a fully paid up member of Worm Liberation.

Imagine then my shock and horror when looking through the 40 colour pictures which illustrate the Macintosh manual — some of them double page spreads — when I found that there is not a Worm in sight. Much worse, there is not a single woman in sight, except possibly for page 148 where at the far end of a colonnade, there is a misty figure that just might be a female in jeans and riding boots — and on the other hand might not be.

Considering that the manual was written by a woman — take a bow, Carol Kaehler — this is a direct and quite obnoxious insult to all women who, for the information of Apple in Cupertino,
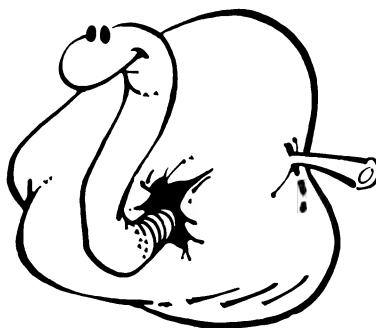
make up half the population of the world, and the more intelligent half at that.

Not only are the illustrations sexist to a degree; they appear to have been specifically selected to give everyone over the age of 25 a feeling of decrepitude. All of the males in the illustrations are plainly young university students who all look as if they are working on the principle "Never trust a person over thirty".

Why aren't there photographs of lined broken old men like David Roman of Apple Australia? Why are there NO women? Is Apple a sexist organisation? It is noticeable that the designer of the manual is not listed. Possibly because an irate customer might castrate him with a rusty razor blade.

And there not a single picture of a Worm. On behalf of the other ugly, elderly Worms of this world I protest. And I wouldn't be at all surprised if Apple headquarters in Cupertino becomes covered in graffitti writ in large letters by irate members of Women's Lib. And the Worms of the world will be right there beside them at the barricades. □
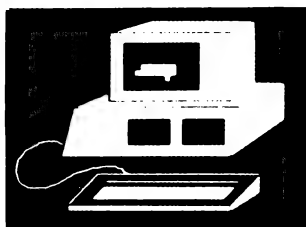
Program a Winner at the Track

by Gareth Powell

I wrote this article originally for the "Weekend Australian". To those select few who already read that august paper, my apologies. However, there is a consolation prize. The "Weekend Australian" charged $2 for the program listing that appears at the end. You get it free.

Dick Francis writes thrillers about horse racing. And damn good they are too.

Francis is writing about something at which he is an expert, for he was a champion steeplechaser for many years. When his books encompass another field apart from horse racing then his research is always meticulous.

In 1981 he wrote "Twice Shy" (still available in paperback from Pan) in which he wrote about the possibility of programming a microcomputer to beat the bookies.

In the book the microcomputer is called a "Grantley", a fictional name. Judging by the segments of programs included it is based on an amalgam of the TRS 80 and the Apple.

"There was provision for scoring for any number of horses in each of more than 800 named races, and in an unknown quantity of unnamed races. Each race had its own set of weightings and very often its own set of questions."

In "Twice Shy" the program works only too well. Sundry nasty gentlemen with firearms come after our hero, only to meet with their comeuppance which is only right and proper in a book of this sort.

But that is fiction.

Is it possible to forecast the winners of horse races on a microcomputer in real life? The answer is a firm and unequivocal "maybe".

In the '60s a betting syndicate called "The Legal Eagles", led by Don Scott and including among its members Clive Evatt Jr, undoubtedly gave Australia's bookies a run for their money.

They used a system which was based on empirical experience, intelligent observation and mathematical probability.

At one stage the syndicate was employing a staff of 12 to keep form cards, film races and place bets.

In 1975 Don Scott stopped being a professional punter and wrote a book called "Winning, An Objective Guide to Successful Punting". The introduction was provided by Bill Waterhouse and "Winning" is possibly still the best book on horse race betting written in Australia.

If the basic hypotheses advanced in this book are true then, in theory, a microcomputer could take the place of that staff of 12 and maintain records which would be a winning computer system.

I became interested in this possibility of a microcomputer based betting system when Graeme Philipson, a computer industry commentator with "The Yankee Group" – an organisation that tells you more about computers than you want to know – lent me a remarkable document.

It was a thesis on the profitability of horse race gambling by Scott Williams, BEc, Dip Rur Acc, who had written it as a "dissertation submitted in partial fulfilment of the requirements for the degree of Master of Economics of the University of New England".

The thesis starts off by disarmingly stating that among its several aims it should let the author "lead a life of ease on the winnings from the implementation of the various systems".

When the thesis was written, the year was 1975, ancient history in microcomputer terms, and the programs were designed for implementation on the University of New England's ICL 1904A computer on strictly rationed time.

This was not the ideal set up but, even so, Scott Williams attempted to answer the question: "Can analytical techniques, typical of those we would expect to use in business management, be used to formulate strategies that can be used for betting on Australian horse races so as to achieve profits comparable with other forms

of investment?"

And the quite startling conclusion that he came to at the end of his thesis was that "race gambling, in even a reasonably unsophisticated manner, is potentially quite considerably more profitable than other forms of common investment".

Being "reasonably unsophisticated" I was entranced at having found a means by which I could cease working for a living and become a rambling, gambling man through the virtues of a system run on a microcomputer.

An Irish mathematical genius had once explained to me that the chances of winning the Irish Sweep were almost precisely the same whether you bought a ticket or not. But this was different. This was using modern technology to break the bank.

The phrase "being reasonably unsophisticated" takes on a new meaning when I tell you that up until that time I had never bet on a horse in my life.

I therefore went into a short-lived betting partnership with Graeme "Phar Lap" Philipson in which we attempted to bring the bookmaking establishment of Australia to its knees through an initial gambling investment of $100. We did not succeed.

The program that we used was sold in the United States under the misleading title of "Win at the Races". It used a mathematical statistical formula called Multiple Regression Analysis Factor for each major racetrack in Australia.

We also modified the program to suit Australian conditions. We were cleaned out on the first day's racing.

There was a good and proper reason for this.

We insisted on placing a bet on every race. There's no racing system in existence which will let you place a bet on every race – and win.

But to get a list out of the program meant that Phar Lap and I worked like maniacs from the early dawn of Saturday morning until race time, punching in the information that was required.

Having gone to all that effort we thought the least we could do was to place a small bet to see what happened. And thus our $100 went down the glug hole.

I tried the system once again in Hong Kong – and it didn't work there either. Again for the same reason. Having expended that much effort in entering in the information, I wanted a bet in order to at least have a chance of seeing a return on effort invested.

I bet $HK1000 and won $HK620. A net loss of $HK380. As a system this leaves something to be desired.

If before that time I had read Don Scott I would have realised that the situation was not a hopeful one. He says: "Final times and sectional times are constantly being fed into dozens of computers around the nation. The computers are given every horse from first to last that are raced at a meeting and are solemnly asked to declare what time each horse ran and what time each horse should have run.

"The following Saturday they have to answer for all of the horses how fast they are going to gallop carrying separate weights over each separate distance that day. And computers have to answer these questions rain, hail or shine. No wonder they often break down under the strain."

Since that time I have learnt a lot about computer forecasting of races.

Although I have totally given up betting – once bitten, twice miserly – I have tried to read every skerrick of information about the subject so that I can now tell you, with authority, that Babbage tried to invent the first computer so that his inamorata, Ada Lovelace, could win at the races.

I have also talked to several professional punters who use computers to direct them in their betting. (It is of interest that most of them seem to have names coined by Damon Runyon.)

From this I have come to some conclusions:

A microcomputer can help you to achieve better results in betting on horse races.

If you insist on betting on every race you will lose.

Certain types of race are difficult, if not impossible, for the computer to analyse. You should not bet on these. Examples are races beyond 2000m, races held in wet conditions, races which are not held at major metropolitan tracks, races which have more than 12 runners, races which

have less than four runners and races in which one or more of the runners have a previous form.

The amount of time and effort that is required to enter all of the necessary information means that, unless you start with a substantial bankroll, you are working for about 27c an hour.

Having said that, I repeat that I believe it is possible to be more successful at the races using a microcomputer than otherwise.

The program we have produced uses some reasonably sophisticated computing techniques incorporating Multiple Regression Analysis. The program will at least let you lose scientifically, if not make a fortune in the first week.

The program has been put together by a troika of talents. The three people concerned are Ethan Dorfmann of Hitachi, who used his considerable programming skills – once the concept had been explained to him.

Second member of the team was Graeme "Phar Lap" Philipson who is a computer industry analyst for "The Yankee Group" and is also, as his nickname suggests, a man who has spent considerable time and effort following the horses.

And lastly I plead guilty to having worked out the concept of the program and having spent a large amount of time, but no money, in putting the various components together and checking the results.

When we completed the first draft of the program, we found that, as it stood, it had two flaws.

The first was that we had gone too far in showing off our programming skills. Nothing exceeds like excess.

After the program had shown the winners in list form it produced a graphic representation on the screen of the horses at the winning post with the first three horses nmed and in their correct relative position with all the jockeys in their right colours.

Which is pretty spectacular.

But doesn't help you to win at the races. It also involves typing in the colours that each jockey is going to wear in each race and this is an added burden which may put you off using the program to its fullest advantage.

The second problem was that we

```
5    CLEAR
10   DIM N$(16),RH(16),B(16),PP(16),BN(16),D(16),P(16),W$(16)
20   FOR I = 1 TO 15: READ B(I): NEXT I
30   HOME
40   INVERSE
45   PRINT "*********************************************"
50   PRINT "***WEEKEND AUSTRALIAN RACING PROGRAM***"
60   PRINT "*********************************************"
65   NORMAL
66   PRINT : PRINT "THE INFORMATION REQUIRED ON 'QUALIFYING RUN' IS PRINTE
     D IN EVERY 'WEEKEND AUSTRALIAN' FOR SATURDAY METROPOLITAN MEETINGS"
67   PRINT : PRINT "A 'QUALIFYING RUN' IS THE LAST RACE RUN BY THE HORSE U
     NDER SIMILAR CONDITIONS"
70   PRINT : PRINT
80   PRINT "     (1) 3 YEAR OLDS": PRINT
90   PRINT "     (2) WELTER": PRINT
100  PRINT "     (3) OPEN": PRINT
110  PRINT "     (4) WEIGHT FOR AGE": PRINT
120  PRINT : PRINT : INPUT "     ENTER RACE TYPE: ";X$
130  X = VAL (X$): IF X < 1 OR X > 4 THEN 30
140  HOME
150  PRINT : INVERSE : HTAB 10: PRINT "*********************"
160  HTAB 10: PRINT "***RACE INFORMATION***"
170  HTAB 10: PRINT "*********************"
175  NORMAL
177  REM
178  REM       INPUT PROCEDURE
179  REM
180  PRINT : PRINT : PRINT
190  INPUT "RACE NUMBER       : ";X$:RN = VAL (X$): PRINT
200  INPUT "DATE OF RACE      : ";DT$: PRINT
210  INPUT "NAME OF TRACK     : ";TR$: PRINT
220  IF X = 2 THEN 250
230  INPUT "DISTANCE          : ";X$:DI = VAL (X$): PRINT
250  INPUT "NUMBER OF HORSES  : ";X$:N = VAL (X$)
260  IF N > 15 THEN PRINT "<< MAXIMUM HORSES = 15 >>": GOTO 250
265  FOR I = 1 TO N
270  HOME
275  INVERSE
280  PRINT "*****************************************"
285  NORMAL
290  PRINT "   RACE ";RN;" AT ";TR$;" ON ";DT$
295  PRINT : INVERSE
300  PRINT "*****************************************"
305  NORMAL
307  D = DI
310  PRINT
320  PRINT "   HORSE NUMBER ";I:PP(I) = I: PRINT
330  INPUT "NAME         : ";N$(I): PRINT
335  IF LEN (N$(I)) > 13 THEN PRINT "<< MAX NAME 13 LETTERS >>": GOTO 3
     30
340  INPUT "WEIGHT          : ";W: PRINT
350  IF W < 50 OR W > 70 THEN PRINT "<< WEIGHT BETWEEN 50 AND 70 >>": GOTO
     340
360  INPUT "BARRIER POSITION : ";P: PRINT
370  IF P < 1 OR P > 15 THEN PRINT "<< ONLY 15 POSITIONS >>": GOTO 360
380  BP = B(P):BN(I) = P
390  INPUT "JOCKEY ALLOWANCE:";JA$:JA = VAL (JA$): PRINT
400  IF JA > 5 THEN PRINT "<< JOCKEY ALLOWANCE MAX = 5 >>": GOTO 390
410  INPUT "WEIGHT ALLOWANCE : ";WA$:WA = VAL (WA$): PRINT
420  IF WA < - 5 OR WA > 5 THEN PRINT "<< WEIGHT ALLOWANCE RANGE -5 TO
     5 >>": GOTO 410
430  GOSUB 1990
431  TC = SC
441  INPUT "WEIGHT IN QUALIFYING RUN : ";W: PRINT
442  INPUT "BEATEN LENGTHS IN QUALIFYING RUN : ";BQ: PRINT
443  INPUT "DISTANCE OF QUALIFYING RUN : ";D: PRINT : INPUT "CORRECT (Y/N
     ) : ";S$: IF S$ = "N" THEN 270
444  GOSUB 1990
450  EP = (W - BQ * 1.5) - SC
470  RH(I) = TC + EP - BP + JA + WA
480  NEXT I
489  REM
490  REM      SORT HANDICAPS
491  REM
500  FOR J = 1 TO N - 1
510  L = J
520  FOR K = J + 1 TO N
530  IF RH(L) > RH(K) THEN 550
540  L = K
550  NEXT K
560  TEMP = RH(L):T1 = PP(L)
570  RH(L) = RH(J):PP(L) = PP(J)
580  RH(J) = TEMP:PP(J) = T1
590  NEXT J
596  GOSUB 1010
600  HOME
605  INVERSE
610  PRINT "*********************************************"
615  NORMAL
620  PRINT "        FINAL RESULTS FOR RACE ";RN
623  PRINT
625  INVERSE
630  PRINT "*********************************************"
635  NORMAL
640  PRINT
650  PRINT "POSITION","HORSE","ODDS"
670  PRINT "============================================"
680  FOR I = 1 TO N
690  PRINT I,N$(PP(I)); SPC( 16 - LEN (N$(PP(I))));W$(I)
```

```
700  NEXT I: PRINT
710  PRINT "============================================"
720  PRINT : INPUT "ANOTHER RACE        (Y/N) : ";Y$: IF Y$ = "Y" THEN
     5
750  END
1000 DATA 0,0,.25,.25,.5,.5,.5,.5,.5,.75,.75,1,1,1,1
1004 REM
1005 REM        ODDS CALCULATION
1006 REM
1010 T = 0
1020 FOR Z = 1 TO N
1025 M = RH(1) - RH(Z)
1030 IF M = 0 THEN D = 1: GOTO 1240
1040 IF M < .5 THEN D = .9: GOTO 1240
1050 IF M < 1. THEN D = .8: GOTO 1240
1060 IF M < 1.5 THEN D = .67: GOTO 1240
1070 IF M < 2 THEN D = .57: GOTO 1240
1080 IF M < 2.5 THEN D = .5: GOTO 1240
1090 IF M < 3 THEN D = .4: GOTO 1240
1100 IF M < 3.5 THEN D = .33: GOTO 1240
1110 IF M < 4 THEN D = .25: GOTO 1240
1120 IF M < 4.5 THEN D = .2: GOTO 1240
1130 IF M < 5 THEN D = .14: GOTO 1240
1140 IF M < 5.5 THEN D = .11: GOTO 1240
1150 IF M < 6 THEN D = .08: GOTO 1240
1160 IF M < 6.5 THEN D = .06: GOTO 1240
1170 IF M < 7 THEN D = .05: GOTO 1240
1180 IF M < 7.5 THEN D = .04: GOTO 1240
1190 IF M < 8.5 THEN D = .03: GOTO 1240
1200 IF M < 9 THEN D = .02: GOTO 1240
1210 IF M < 9.5 THEN D = .01: GOTO 1240
1220 IF M > 9.5 THEN D = 0
1240 D(Z) = D
1245 T = T + D(Z)
1250 NEXT Z
1300 FOR Q = 1 TO N
1310 P(Q) = D(Q) / T * 80
1320 P(Q) = INT (P(Q) + .5)
1330 P = P(Q)
1350 IF P > = 75 THEN W$ = "1/3"
1360 IF P < 75 AND P > = 73 THEN W$ = "4/11"
1370 IF P < 73 AND P > = 71 THEN W$ = "2/5"
1380 IF P < 71 AND P > = 68 THEN W$ = "4/9"
1390 IF P < 68 AND P > = 66 THEN W$ = "1/2"
1400 IF P < 66 AND P > = 65 THEN W$ = "8/15"
1410 IF P < 65 AND P > = 63 THEN W$ = "4/7"
1420 IF P < 63 AND P > = 61 THEN W$ = "8/13"
1430 IF P < 61 AND P > = 59 THEN W$ = "4/6"
1440 IF P < 59 AND P > = 57 THEN W$ = "8/11"
1445 IF P < 57 AND P > = 55 THEN W$ = "4/5"
1450 IF P < 55 AND P > = 52 THEN W$ = "9/10"
1455 IF P < 52 AND P > = 49 THEN W$ = "EVENS"
1460 IF P < 49 AND P > = 46 THEN W$ = "10/9"
1470 IF P < 46 AND P > = 44 THEN W$ = "5/4"
1480 IF P < 44 AND P > = 42 THEN W$ = "11/8"
1490 IF P < 42 AND P > = 40 THEN W$ = "6/4"
1500 IF P < 40 AND P > = 38 THEN W$ = "13/8"
1510 IF P < 38 AND P > = 36 THEN W$ = "7/4"
1520 IF P < 36 AND P > = 35 THEN W$ = "15/8"
1530 IF P < 35 AND P > = 33 THEN W$ = "2/1"
1540 IF P < 33 AND P > = 30 THEN W$ = "9/4"
1550 IF P < 30 AND P > = 28 THEN W$ = "5/2"
1560 IF P < 28 AND P > = 26 THEN W$ = "11/4"
1570 IF P < 26 AND P > = 25 THEN W$ = "3/1"
1580 IF P < 25 AND P > = 23 THEN W$ = "13/4"
1590 IF P < 23 AND P > = 22 THEN W$ = "7/2"
1600 IF P < 22 AND P > = 21 THEN W$ = "15/4"
1610 IF P < 21 AND P > = 20 THEN W$ = "4/1"
1620 IF P < 20 AND P > = 18 THEN W$ = "9/2"
1630 IF P < 18 AND P > = 17 THEN W$ = "5/1"
1640 IF P < 17 AND P > = 15 THEN W$ = "11/2"
1650 IF P < 15 AND P > = 14 THEN W$ = "6/1"
1660 IF P < 14 AND P > = 13 THEN W$ = "7/1"
1670 IF P < 13 AND P > = 12 THEN W$ = "15/2"
1680 IF P < 12 AND P > = 11 THEN W$ = "8/1"
1690 IF P < 11 AND P > = 10 THEN W$ = "9/1"
1700 IF P < 10 AND P > = 9 THEN W$ = "10/1"
1710 IF P < 9 AND P > = 8 THEN W$ = "12/1"
1720 IF P < 8 AND P > = 7 THEN W$ = "14/1"
1730 IF P < 7 AND P > = 6 THEN W$ = "16/1"
1740 IF P < 6 AND P > = 5 THEN W$ = "20/1"
1750 IF P < 5 AND P > = 4 THEN W$ = "25/1"
1760 IF P < 4 AND P > = 3 THEN W$ = "30/1"
1770 IF P < 3 AND P > = 2.5 THEN W$ = "40/1"
1780 IF P < 2.5 AND P > = 2 THEN W$ = "50/1"
1790 IF P < 2 AND P > = 1.5 THEN W$ = "60/1"
1800 IF P < 1.5 AND P > = 1 THEN W$ = "100/1"
1810 IF P < 1 THEN W$ = "100/1 +"
1820 W$(Q) = W$
1830 NEXT Q
1840 RETURN
1990 REM
1991 REM        REGRESSION COEFFICIENTS
1992 REM
2000 IF X = 1 THEN SC = - 270.94 + W * 8.8034 + D * .07268 - W * D * .0
     0085 - D ^ 2 * .000008 - W ^ 2 * .05955
2010 IF X = 2 THEN SC = - 13 + W * 1.167
2020 IF X = 3 THEN SC = - 3.2 + W * 1.156 - D * .009 + D ^ 2 * .000002
2030 IF X = 4 THEN SC = 252.14 - W * 9.688 + D * .003 + W ^ 2 * .1069
2040 RETURN
```

had made no allowance for the interested punter to use experience and judgement. The reslut was a cold sterile program that acted as if it knew it all. Computers don't know it all. They just think they do.

We decided that there were two factors that could be optionally entered by the punter on the basis of experience and judgement.

The first factor is a jockey allowance; the second a weight allowance. We'll come to these in a moment. Note that these value judgements are not absolutely necessary. We just felt that you might like to have some control over your own destiny.

Once you start the program the title page appears you are invited to enter the type of race involved:

1. Three year olds
2. Welter
3. Open
4. Weight for age

Having entered the type of race you are working on you are confronted with another screen asking for race information.

1. Date of race
2. Name of track
3. Race number
4. Distance
5. Number of horses

The next screen confirms the information you have entered with a listing across the top showing which race it is, where and at what date. You then have to enter the following information.

1. Horse number
2. Horse name
3. Weight carried
4. Position at barrier
5. Weight allowance

(If you press RETURN at this point the value given will automatically zero. However, if you fancy yourself in making value judgements regarding horses in a particular race, go for your life. What you are then doing is effectively handicapping the horses. This requires judgement and expertise and we have a sneaky feeling that the end result is going to favour the punter who ignores this and just presses return. Time will tell.)

6. Jockey allowance

(Again, you can ignore this because it is another value judgement. An apprentice first time up might justifiably be given a rating of −1.

One of the seasons top twenty jockeys, on the other hand, might be given an allowance of +1. Figures in excess of this either was will skew the results to such an extent that they become meaningless.

Once you have entered these vital statistics you will see almost immediately on your screen a prediction of the results.

From left to right will be given predicted position, name of horse, barrier drawn and expected odds.

In this the first of a series we have checked out our own Multiple Regression Factors and entered them in as constants.

Which means that this program is not designed to update itself as you go along by adjusting the MRA factor after the results of each race have been obtained and entered in comparison with the prediction.

We can and will do this at a later date but the amount of work that you, the punter, will have to do then will increase enormously. In order to arrive at a satisfactory weighting you will need to enter the historical record of each horse.

This is a time consuming and expensive buseiness. Either you spend the wee hours before each race day frantically going through the form books and entering the last five significant races with all relevant details (barrier position, weight carried, race track, race length, rating of jockey time at the Leger, completed race time) or you retain all of this information on a 10 Megabyte hard disk so that you eventually build up a file of information on every galloper in Australia.

Historical record

We are following the second method after this program has been running for eight weeks we may issue new weightings.

You will easily be able to enter these into the program if significant changes have occured.

We are also working on a program that can download through the telephone current information from one of the news services so that the keying in of the information will not be necessary. We will keep you informed.

The most important information

that this current program will give you is "forecast odds". It is here that the intelligent and patient punter will make money.

If the projected odds are in line with those being quoted then it is possible that the other forecasters are using a similar method to ours. (Laughable though you may find this we found we were uncannily close on three trial runs).

What you will be looking for is an "overlay". That is where the program says that the winner's odds shoud be, say, 4/1 and the odds predicted as shown in your newspaper are 12/1. This is a profit potential situation.

When the results come up on your screen or printer if you have one look first for the "overlays" and mark them for closer study.

Reject any race where the difference in odds between the first three horses is insignificant. The bigger the spread, the bigger your chances of success.

1. Do not bet on a race where there are four runners or less.

2. Do not bet on a race where there are more than fifteen runners.

3. Do not bet on anything except the major metropolitan meetings.

4. Do not bet on a race where one or more of the horses has no record whatsoever.

5. Do not bet if the track is wet and heavy.

6. Do not bet unless you can expect an "overlay" on the odds.

7. Do not bet if the predicted positions of the first three horses is close. The level of acceptable closeness is a value judgement you can make.

8. Do not bet unless you can well afford the money. The chances of you making a steady return on your money are not high unless you have a comfortable bankroll and yu are a disciplined and motivated punter.
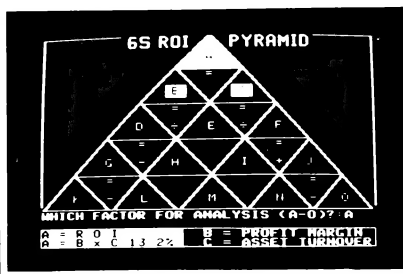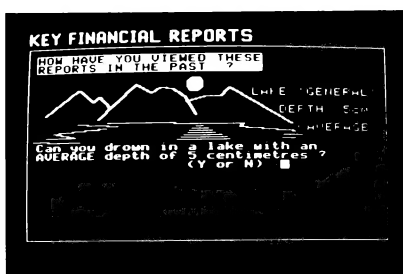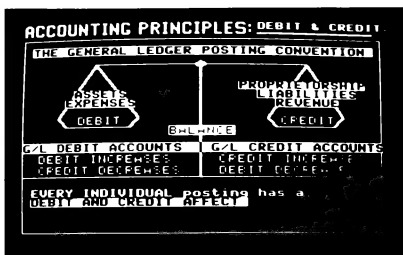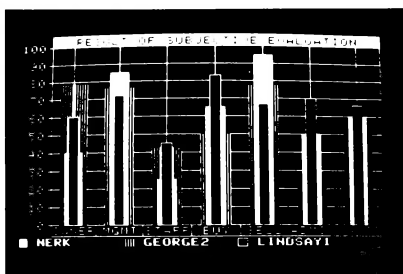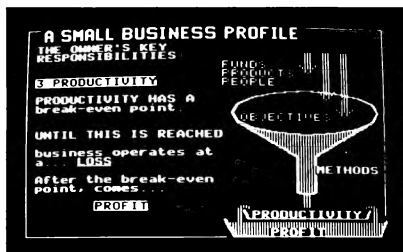
9. Do not bet. Ethan Dorfmann and I do not bet. Phar Lap Philipson does. Ethan Dorfmann and I drive new cars. Phar Lap Philipson has a Morris Minor which has lurched past its majority and is a static monument to English engineering.

The program as it is written has been tested and works. But we can make no guarantees as to whether you will make a fortune beating the bookies.

May the force be with you.   □

Introducing a Teacher You Can Give to Your Apple.

For the first time in Australia, quality educational courseware is available for your Apple IIE or your Apple II Plus. Control Data PLATO* is proud to announce that from December 1st, new lessons from the vast PLATO computer based education library will be available for your Apple micro.

Control Data PLATO computer-based education, recognised as a world leader in educational software, is a highly interactive system of self-paced, one-to-one instruction. Developed over 20 years, it has been thoroughly tested and proven in business, industry, and the academic community. PLATO lessons are designed and programmed to motivate and direct students as well as helping them develop a sense of accomplishment and personal progress.

For example, the course titled Computer Literacy provides the foundation for simple programming as well as giving a brief introduction to the uses of computers in to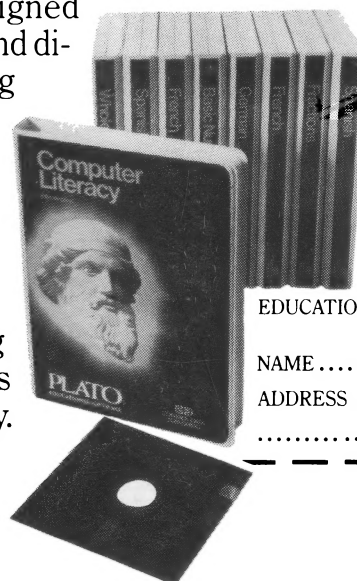day's society. There are lessons in building vocabulary in French, German and Spanish as well as lessons in Physics, Fractions, Decimals and Basic Number Facts.

Today, more than ever before, all members of society need to learn. PLATO computer-based education is designed and paced for the individual. This means easier learning with infinite patience and personalization.

As traditional education methods become progressively more costly and less efficient you owe it to yourself to bring the power of PLATO learning to your Apple.

If you would like to be on our mailing list for future information on PLATO Courseware for your Apple, please fill in and mail the coupon below.